This document is downloaded from DR-NTU (https://dr.ntu.edu.sg)
Nanyang Technological University, Singapore.

Low-voltage, low-power CMOS arithmetic circuits for energy efficient VLSI applications

Gu, Jiangmin

2005

Gu, J. (2005). Low-voltage, low-power CMOS arithmetic circuits for energy efficient VLSI applications. Master’ s thesis, Nanyang Technological University, Singapore.

https://hdl.handle.net/10356/4315

https://doi.org/10.32657/10356/4315

Nanyang Technological University

Downloaded on 21 Aug 2022 01:38:52 SGT

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library
0365158

Low-Voltage, Low-Power CMOS Arithmetic Circuits for Energy Efficient VLSI Applications

Gu Jiangmin

NANYANG TECHNOLOGICAL UNIVERSITY
20 JUL 2005
LIBRARY

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfilment of the requirement for the degree of
Master of Engineering

2005

TK
7871.99
M44
G896
2005

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Acknowledgements

I wish to thank my supervisor, Assistant Professor Chang Chip Hong, for his efficient supervision, active academic motivation and easy-going communication throughout the research program. In particular, I wish to thank him for teaching me the philosophies of research and intangible skills, which are the most invaluable knowledge I have acquired in the research program.

I would also like to thank Associate Professor Yeo Kiat Seng for the discussion and great help in the development of the algorithm, practical layout and the fabrication of the scalar product macrocell.

I would like to thank Ms Zhang Mingyan and Ms He Yajuan for the long run friendship, thorough discussion, hitting suggestions and invaluable help pertaining to my research. I would like to thank Ms Gong Yuanyuan and Mr. Shi Lei for their tedious but indispensable work to speed up the research schedule. I would like to thank Mr. Xu Pengfei as well for both the firm friendship and kind help pertaining to the research.

To the staff and other students in Integrated System Research Laboratory, IC Design Laboratories 1 and 2, Center for Integrated Circuits and Systems and Center for High Performance Embedded Systems, I wish to convey my appreciation to all of them for their kind and friendly assistance.

I would like to express my gratitude to my parents for their consistent support and encouragement during all periods of my study.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Summary

We present in this thesis several algorithms and designs, and their IC implementation of low-voltage, low-power CMOS arithmetic circuits for energy efficient VLSI applications. The design methodologies span from circuit level, architecture level to algorithm level.

In the circuit level, we study basic gates including the XOR gate and propose a pass-transistor logic design style XOR-XNOR gate used to efficiently generate XOR and XNOR functions simultaneously and another gate to generate good drivability carry output in a novel complementary CMOS style circuit with regular layout. Based on these two gates, we propose two novel 1-bit adder cells, Hybrid-1 and Hybrid-2. The structures of 4-2 and 5-2 compressors, which are the main building blocks of multipliers, are analyzed and different CMOS logic style circuit implementations of their constituent modules are explored. We proposed a new 4-2 compressor using the novel XOR-XNOR gate and a novel 5-2 compressor architecture of 4Δ delay, both of which are able to function down to 0.6V, and feature high speed and low power characteristics.

In the algorithm and architecture level, we present a new VLSI circuit design algorithm for scalar product evaluation. It produces a novel full bit-parallel architecture of scalar product macrocell featuring a low interconnect complexity, improved power efficiency and highly efficient VLSI area utilization. More importantly, the layout regularity and scalability enhance its performance superiority in deep submicron regime well above conventional VLSI design of vector processing unit for scalar product computation. The chip is fabricated on Chartered Semiconductor Manufacturing 0.18μm CMOS technology. To test the scalar product macrocell, several auxiliary circuits including a unique delay detection circuit are proposed to enable accurate delay measurement for the core with the constraint of limited IO pins.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

We exploit the redundant binary system and propose another new redundant binary multiplier based on the newly developed Covalent Redundant Binary Booth Encoding (CRBBE) algorithm. This algorithm fully exploits the characteristics of the Booth encoded numbers to overcome the problem of generating hard multiples and achieves a compatible reduction of Redundant Binary (RB) partial product without inducing any correction vector. Compared with the existing Redundant Binary Signed Digit (RBSD) multiplier, which is also used to address the hard multiple problem, the proposed CRBBE multiplier has only half the number of partial products to be summed if the same radix Booth encoders are used. Conversely, for the same number of RB partial products, the CRBBE multiplier is much simpler, thus less power consuming and faster than the RBSD multiplier and the RB partial product generation with normal Booth encoding (NBE-RBPPG) multiplier, which uses many high fan-in gates.

To optimize and simulate the arithmetic cells, we first propose an optimization procedure to size the transistor of all arithmetic cells in order to perform a fair comparison because the transistor sizing for optimal performance is technology dependent. The optimization procedure takes sweeping method to search for the optimized transistor sizes in iterations, which has been proven to be a convergent algorithm. We also propose reasonably simulation structures to compare the basic arithmetic cells, such as full adders, 4-2 and 5-2 compressors, in an environment realistic to their actual deployment in the most frequently used parallel multiplier structure.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Contents

ACKNOWLEDGEMENTS ..... **I**

SUMMARY ..... **III**

CONTENTS ..... **V**

LIST OF FIGURES ..... **VII**

LIST OF TABLES ..... **X**

LIST OF ABBREVIATIONS ..... **XI**

CHAPTER 1 INTRODUCTION **1**

1.1 MOTIVATION 1
1.2 OBJECTIVE 3
1.3 RESEARCH ORIGINALITY 4
1.4 ORGANIZATION 6

CHAPTER 2 LITERATURE REVIEW **9**

2.1 LOW POWER TECHNIQUES FOR DIGITAL CIRCUIT DESIGN 9
2.1.1 Overview of power consumption of CMOS circuit *11*
2.1.2 Low power design techniques *14*
2.2 DESIGN OF XOR AND XNOR CELLS 19
2.3 DESIGN OF FULL ADDERS 30
2.4 DIGITAL MULTIPLIER ARCHITECTURES 35
2.4.1 Normal binary multipliers *35*
2.4.2 Redundant binary multipliers *39*

CHAPTER 3 DESIGN AND PERFORMANCE EVALUATION OF LOW VOLTAGE LOW POWER FULL ADDER CELLS **42**

3.1 INTRODUCTION 42
3.2 REVIEW OF EXISTING FULL ADDER CELLS 44
3.3 FULL ADDER ARCHITECTURE AND ITS BUILDING BLOCKS 47
3.3.1 XOR/XNOR module *48*
3.3.2 XOR module for sum output *51*
3.3.3 Carry generator module *51*
3.3.4 Circuit Structure of Hybrid 1 and Hybrid 2 ..... *54*
3.4 SIMULATION RESULTS 55
3.4.1 Simulation environment *55*
3.4.2 Transistor sizing optimization *59*
3.4.3 Simulation results and analysis *65*
3.5 SUMMARY 74

CHAPTER 4 LOW VOLTAGE, LOW POWER 4-2 AND 5-2 COMPRESSORS FOR FAST ARITHMETIC CIRCUITS **76**

4.1 INTRODUCTION 76
4.2 4-2 COMPRESSOR 77
4.3 5-2 COMPRESSOR 81
4.3.1 5-2 COMPRESSOR ARCHITECTURES *81*
4.3.2 Circuit of the proposed 5-2 compressor architectures *87*
4.4 SIMULATION RESULTS 88
4.4.1 Simulation environment *88*
4.4.2 Simulation results of 4-2 compressors *90*

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

4.4.3 Simulation Results of 5-2 Compressors *95*
4.5 SUMMARY 108

CHAPTER 5 AN AREA AND ENERGY EFFICIENT IP CORE FOR SCALAR PRODUCT COMPUTATION 110

5.1 INTRODUCTION 110
5.2 ALGORITHM OF SCALAR PRODUCT 113
5.3 ARCHITECTURAL DESIGN OF THE SCALAR PRODUCT CORE 120
5.4 FLOOR PLANNING OF THE SCALAR PRODUCT CORE 123
5.5 DELAY ESTIMATION BASED ON FLOOR PLANNING 130
5.6 THE SCALAR PRODUCT CORE LAYOUT IN CSM 0.18MM TECHNOLOGY 133
5.7 DESIGN OF THE ON-CHIP AUXILIARY CIRCUIT 136
5.8 SIMULATION RESULTS 140
5.9 SUMMARY 144

CHAPTER 6 COVALENT REDUNDANT BINARY BOOTH ENCODED MULTIPLIER 145

6.1 INTRODUCTION 145
6.2 REDUNDANT BINARY CODING AND RB MULTIPLIER STRUCTURES 147
6.3 EXISTING BOOTH ENCODING AND PARTIAL PRODUCT GENERATION METHODS FOR RB MULTIPLICATION 149
6.3.1 Normal Booth encoding *150*
6.3.2 NBE-RBPPG: RB partial product generation with normal Booth encoding *153*
6.3.3 RBSD Booth encoding and partial product generation *155*
6.4 CRBBE: PROPOSED BOOTH ENCODING AND PARTIAL PRODUCT GENERATION ALGORITHMS FOR RB MULTIPLICATION 160
6.4.1 CRBBE-1: the Booth-1 encoding for RB multiplication *160*
6.4.2 CRBBE-2: the Booth-2 encoding for RB multiplication *162*
6.4.3 CRBBE-1.5: A mixture of CRBBE-1 and CRBBE-2 ..... *164*
6.5 PORTING OF CRBBE ALGORITHM TO RB MULTIPLIER ARCHITECTURE 165
6.5.1 Circuit implementation of CRBBE encoders and RB PPGs *165*
6.5.2 Redundant Binary Adders *172*
6.5.3 RB to NB Conversion *175*
6.5.4 Proposed CRBBE 54 × 54-bit multiplier *177*
6.6 SIMULATION RESULTS 179
6.7 SUMMARY 180

CHAPTER 7 CONCLUSIONS AND FUTURE RESEARCH 182

7.1 CONCLUSIONS 182
7.2 FUTURE RESEARCH 185
7.2.1 Exploitation of the proposed full adders in fixed multiplierless digital filter implementations *185*
7.2.2 Exploitation of 4-2 and 5-2 compressors in the design of multiply-accumulator *186*
7.2.3 Integration of the scalar product macrocell in special purpose digital signal processors *186*
7.2.4 Design, floor planning and layout of double precision floating point multiplier with the proposed CRBBE algorithm *187*
7.2.5 Scalable integer multiplier using the proposed CRBBE algorithm *187*
7.2.6 Digital filters and correlators using the proposed CRBBE algorithm *188*
7.3 DESIGN CHALLENGES IN FUTURE ULTRA DEEP SUBMICRON TECHNOLOGIES 188

AUTHOR'S PUBLICATIONS 190

BIBLIOGRAPHY 192

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

List of Figures

FIGURE 2.1 CAPACITIVE LOAD MODEL 11
FIGURE 2.2 POWER OPTIMIZATION HIERARCHY 15
FIGURE 2.3 CLASSICAL CMOS XOR-XNOR GATE 20
FIGURE 2.4 MIRROR CMOS XOR AND XNOR GATES 21
FIGURE 2.5 WANG'S CROSS-BACK XOR AND XNOR GATES 21
FIGURE 2.6 MODIFIED CROSS-BACK XOR AND XNOR GATES 22
FIGURE 2.7 CMOS+ XOR AND XNOR GATES 23
FIGURE 2.8 MODIFIED CMOS+ XOR AND XNOR GATES 23
FIGURE 2.9 CPL XOR AND XNOR GATES 24
FIGURE 2.10 LATCHED CPL XOR AND XNOR GATES 24
FIGURE 2.11 DPL XOR AND XNOR GATES 25
FIGURE 2.12 TRANSMISSION GATE XOR AND XNOR GATES 25
FIGURE 2.13 PASS TRANSISTOR XOR AND XNOR GATES 26
FIGURE 2.14 POWERLESS-XOR AND GROUNDLESS-XNOR GATES 26
FIGURE 2.15 DRIVABILITY ENHANCEMENTS FOR XOR AND XNOR GATES 27
FIGURE 2.16 MODIFIED CPL XOR-XNOR GATE 27
FIGURE 2.17 FEEDBACK XOR-XNOR GATE 28
FIGURE 2.18 CHENG'S XOR-XNOR GATE 28
FIGURE 2.19 MIRROR CMOS 3-INPUT XOR GATE 29
FIGURE 2.20 FANG'S 3-INPUT XOR GATE 30
FIGURE 2.21 CLASSICAL CMOS FULL ADDER 31
FIGURE 2.22 MIRROR CMOS FULL ADDER 31
FIGURE 2.23 FAST CARRY CMOS FULL ADDER 32
FIGURE 2.24 MODIFIED CPL FULL ADDER 32
FIGURE 2.25 LATCHED CPL FULL ADDER 33
FIGURE 2.26 STRUCTURE OF FULL ADDER 34
FIGURE 2.27 CLASSIFICATION OF MULTIPLIERS 36

FIGURE 3.1 BLOCK DIAGRAM OF FULL ADDER 44
FIGURE 3.2 EXISTING FULL ADDER CELLS 46
FIGURE 3.3 THREE-MODULE STRUCTURED FULL ADDER 47
FIGURE 3.4 CIRCUIT IMPLEMENTATIONS OF XOR/XNOR MODULE 49
FIGURE 3.5 SLOW STEP OUTPUT TRANSITIONS WHEN AB CHANGE FROM 01, 10 TO 00 FOR THE XOR/XNOR CIRCUIT OF FIG. 3.4(D) AT SUPPLY VOLTAGE OF 0.8V 50
FIGURE 3.6 XOR MODULE CIRCUITS FOR SUM OUTPUT 51
FIGURE 3.7 IMPLEMENTATIONS OF THE CARRY GENERATOR MODULE USING MUX 52
FIGURE 3.8 PROPOSED CARRY GENERATOR MODULE FOR FULL ADDER 53
FIGURE 3.9 PROPOSED 1-BIT FULL ADDER CIRCUIT - HYBRID 1 54
FIGURE 3.10 PROPOSED 1-BIT FULL ADDER CIRCUIT- HYBRID 2 55
FIGURE 3.11 SIMULATION SETUP SUGGESTED IN [SHA02] 56
FIGURE 3.12 PROPOSED SIMULATION SETUP 57
FIGURE 3.13 WAVEFORM SNAPSHOTS OF THE CIRCUITS WITH ATTACHED BUFFERS (1.8V, 100MHZ) 58
FIGURE 3.14 WAVEFORM SNAPSHOTS OF THE CIRCUITS WITHOUT ATTACHED BUFFERS (1.8V, 100MHZ) 59
FIGURE 3.15 OPTIMIZATION FLOWCHART 61
FIGURE 3.16 POWER, DELAY AND POWER-DELAY-PRODUCT COMPARISON OF FULL ADDER CELLS 68
FIGURE 3.17 NORMALIZED DELAY DIFFERENCE FACTORS BETWEEN HYBRID-2 AND C-CMOS 70
FIGURE 3.18 COMPARISON OF OUTPUT WAVEFORMS BETWEEN HYBRID-2 AND C-CMOS 70
FIGURE 3.19 LAYOUT OF THE HYBRID-1 CELL 72
FIGURE 3.20 LAYOUT OF THE HYBRID-2 CELL 72

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

FIGURE 4.1 *4-2 COMPRESSOR* 78
FIGURE 4.2 *CONVENTIONAL 4-2 COMPRESSOR - 4Δ* 78
FIGURE 4.3 *LOGIC LEVEL OPTIMIZED CMOS 4-2 COMPRESSOR* 79
FIGURE 4.4 *LOGIC DECOMPOSITION OF 4-2 COMPRESSOR - 3Δ* 80
FIGURE 4.5 *LAYOUT OF THE NEW 4-2 COMPRESSOR USING THE PROPOSED XOR-XNOR CELL* 81
FIGURE 4.6 *5-2 COMPRESSOR* 82
FIGURE 4.7 *5-2 COMPRESSOR BASED ON CASCADED FULL ADDERS - 6Δ* 82
FIGURE 4.8 *5-2 COMPRESSOR ARCHITECTURE - 5Δ* 83
FIGURE 4.9 *5-2 COMPRESSOR ARCHITECTURE - 4Δ* 84
FIGURE 4.10 *PROPOSED 5-2 COMPRESSOR ARCHITECTURE - 4Δ* 86
FIGURE 4.11 *IMPLEMENTATIONS OF THE CGEN1 MODULE* 87
FIGURE 4.12 *LAYOUT OF THE PROPOSED 5-2 COMPRESSOR* 88
FIGURE 4.13 *SIMULATION ENVIRONMENTS* 89
FIGURE 4.14 *PERFORMANCES OF 4-2 COMPRESSORS (DESIGNS 2 – 6)* 93
FIGURE 4.15 *PERFORMANCES OF 4-2 COMPRESSORS (DESIGNS 1, 3, 6 – 9)* 95
FIGURE 4.16 *PERFORMANCES OF 5-2 COMPRESSORS (DESIGNS 1 – 5)* 99
FIGURE 4.17 *PERFORMANCES OF 5-2 COMPRESSORS (DESIGNS 6 – 9)* 101
FIGURE 4.18 *PERFORMANCES OF 5-2 COMPRESSORS (DESIGNS 10 – 13)* 102
FIGURE 4.19 *PERFORMANCES OF DIFFERENT 5-2 COMPRESSOR ARCHITECTURES WITH BBB CONFIGURATION (DESIGNS 1, 6, 10)* 104
FIGURE 4.20 *PERFORMANCES OF DIFFERENT 5-2 COMPRESSOR ARCHITECTURES WITH EBB CONFIGURATION (DESIGNS 2, 7, 11)* 105
FIGURE 4.21 *PERFORMANCES OF DIFFERENT 5-2 COMPRESSOR ARCHITECTURES WITH HYBRID CONFIGURATION (DESIGNS 3, 8, 12)* 107
FIGURE 4.22 *PERFORMANCES OF DIFFERENT 5-2 COMPRESSOR ARCHITECTURES OF DUAL-RAIL LOGIC CONFIGURATION (DESIGNS 4, 5, 9, 13)* 108

FIGURE 5.1 *TRADITIONAL ALGORITHM FOR VLSI DESIGN OF SCALAR PRODUCT IP CORE* 115
FIGURE 5.2 *STORED CARRY FORMAT* 118
FIGURE 5.3 *PROPOSED ALGORITHM FOR VLSI DESIGN OF VECTOR MULTIPLIER* 119
FIGURE 5.4 *PROPOSED ARCHITECTURE FOR THE SCALAR PRODUCT CORE* 120
FIGURE 5.5 *RECTANGULAR STRUCTURE OF PPA* 121
FIGURE 5.6 *WALLACE TREE STRUCTURE OF PPA* 121
FIGURE 5.7 *PARALLELOGRAM STRUCTURE OF VA* 122
FIGURE 5.8 *WALLACE TREE STRUCTURE OF VA* 123
FIGURE 5.9 *FLOOR PLANNING OF PPA WITH PPG* 124
FIGURE 5.10 *FLOOR PLANNING OF THE PROPOSED BIT-PARALLEL SCALAR PRODUCT CORE* 126
FIGURE 5.11 *FLOOR-PLANNING OF NORMAL 16×16-BIT MULTIPLIER* 127
FIGURE 5.12 *FLOOR PLANNING OF THE CONVENTIONAL SCALAR PRODUCT MACROCELL* 129
FIGURE 5.13 *LAYOUT OF THE PARTIAL PRODUCT ACCUMULATOR* 133
FIGURE 5.14 *LAYOUT OF THE SCALAR PRODUCT MACROCELL* 134
FIGURE 5.15 *LOCAL AND GLOBAL POWER GROUND LINES ARRANGEMENT* 135
FIGURE 5.16 *INPUT REGISTERS* 137
FIGURE 5.17 *THE 36-INPUT AND GATE* 138
FIGURE 5.18 *DELAY DETECTION CIRCUIT* 139
FIGURE 5.19 *LAYOUT OF DELAY DETECTION CIRCUIT* 139
FIGURE 5.20 *COMPARISON OF POWER DISSIPATION* (MW/10MHZ) 142
FIGURE 5.21 *COMPARISON OF WORST-CASE DELAY* (NS) 143
FIGURE 5.22 *COMPARISON OF POWER EFFICIENCY* (PJ/10MHZ) 143

FIGURE 6.1 *STRUCTURE OF BOOTH DIGITAL MULTIPLIER* 149
FIGURE 6.2 *NBE-RBPPG BOOTH ENCODER* 155
FIGURE 6.3 *ONE DIGIT REDUNDANT BINARY PARTIAL PRODUCT GENERATOR FOR NBE-RBPPG* 155
FIGURE 6.4 *RB ADDER FOR GENERATING THE K-TH RB DIGIT OF THE 5M HARD MULTIPLE* 157

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

FIGURE 6.5 *RBSD BOOTH-2 ENCODER AND RBPP GENERATOR* 157
FIGURE 6.6 *RBSD BOOTH-3 ENCODER AND RBPP GENERATOR* 158
FIGURE 6.7 *RBSD BOOTH-4 ENCODER AND RBPP GENERATOR* 159
FIGURE 6.8 *CBEBE-1 ENCODER* 166
FIGURE 6.9 *CRBBE-1 RB PARTIAL PRODUCT GENERATOR* 167
FIGURE 6.10 *CRBBE-1.5 ENCODER* 167
FIGURE 6.11 *CRBBE-1.5 RB PARTIAL PRODUCT GENERATOR* 168
FIGURE 6.12 *CRBBE-2 ENCODERS* 170
FIGURE 6.13 *RB PARTIAL PRODUCT GENERATOR FOR CRBBE-2* 171
FIGURE 6.14 *SCHEMATIC DIAGRAMS OF DIFFERENT RBAS* 174
FIGURE 6.15 *CIRCUIT TO INHIBIT (1,1) INPUT FOR POSITIVE-NEGATIVE CODING* 175
FIGURE 6.16 *CIRCUIT IMPLEMENTATION OF RBA4* 175
FIGURE 6.17 *CIRCUIT IMPLEMENTATION OF RB-NB CONVERTER* 176
FIGURE 6.18 *PROPOSED CRBBE 54×54-BIT RB MULTIPLIER* 177
FIGURE 6.19 *NBE-RBPPG 54×54-BIT RB MULTIPLIER* 178
FIGURE 6.20 *RBSD 54×54-BIT RB MULTIPLIER* 179

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

List of Tables

TABLE 3.1 TRANSISTOR SIZES (μM) OF FULL ADDERS OPTIMIZED FOR POWER-DELAY PRODUCT 63
TABLE 3.2 POWER, DELAY AND POWER-DELAY-PRODUCT COMPARISON OF FULL ADDER CELLS 66
TABLE 3.3 THE SILICON AREA OF THE FULL ADDERS 73
TABLE 3.4 THE PRE- AND POST-LAYOUT SIMULATION RESULTS OF THE HYBRID-1 AND HYBRID-2 CELLS 74

TABLE 4.1 CONFIGURATIONS OF THE SIMULATED 4-2 COMPRESSORS 91
TABLE 4.2 COMPARISON OF DELAY (NS) OF 4-2 COMPRESSORS 91
TABLE 4.3 COMPARISON OF POWER (μW) OF 4-2 COMPRESSORS 91
TABLE 4.4 COMPARISON OF POWER EFFICIENCY (FJ) OF 4-2 COMPRESSORS 92
TABLE 4.5 CONFIGURATIONS OF THE SIMULATED 5-2 COMPRESSORS 96
TABLE 4.6 COMPARISON OF DELAY (NS) OF 5-2 COMPRESSORS 97
TABLE 4.7 COMPARISON OF POWER (μW) OF 5-2 COMPRESSORS 97
TABLE 4.8 COMPARISON OF POWER EFFICIENCY (FJ) OF 5-2 COMPRESSORS 98

TABLE 5.1 THE SIZES OF DIFFERENT LEAF CELLS (ON CSM 0.18μM CMOS PROCESS) 124
TABLE 5.2 FLOOR PLANNING CHARACTERISTICS OF THE TWO ARCHITECTURES 130
TABLE 5.3 ESTIMATION OF THE WORST CASE DELAY 132
TABLE 5.4 COMPARISON OF POWER, DELAY, AND POWER EFFICIENCY 141

TABLE 6.1 SIGN-MAGNITUDE CODING 147
TABLE 6.2 POSITIVE-NEGATIVE CODING 147
TABLE 6.3 POSITIVE-NEGATIVE COMPLEMENT CODING 148
TABLE 6.4 BOOTH-1 ENCODING 150
TABLE 6.5 BOOTH-2 ENCODING 151
TABLE 6.6 BOOTH-3 ENCODING 151
TABLE 6.7 BOOTH-4 ENCODING 152
TABLE 6.8 RBSD BOOTH-3 ENCODING 156
TABLE 6.9 RBSD BOOTH-4 ENCODING 156
TABLE 6.10 PERMISSIBLE PAIRS OF CONTIGUOUS DIGITS (D_{i+1} D_i) IN BOOTH-1 ENCODED NUMBER 160
TABLE 6.11 PROPOSED COVALENT RB BOOTH-1 ENCODED DUPLET (D_{i+1} D_i) 161
TABLE 6.12 PERMISSIBLE PAIRS OF CONTIGUOUS DIGITS (D_{i+1} D_i) IN BOOTH-2 ENCODED NUMBER 162
TABLE 6.13 PROPOSED COVALENT RB BOOTH-2 ENCODED DUPLETS ($D_{i+1}D_i$) 163
TABLE 6.14 PERMISSIBLE PAIRS OF CONTIGUOUS DIGITS (D_{i+1} D_i) IN COMPOSITE BOOTH-1 AND BOOTH-2 ENCODED NUMBER 164
TABLE 6.15 PROPOSED COVALENT RB BOOTH-1.5 ENCODED DUPLETS (D_{i+1} D_i) 164
TABLE 6.16 CARRY-FREE ADDING RULES FOR RBA 173
TABLE 6.17 SIMULATION RESULTS OF RB 54×54-BIT MULTIPLIERS 180

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

List of Abbreviations

ALU	Arithmetic Logic Unit
ASIC	Application Specific Integrated Circuit
BEPPG	Booth Encoding and Partial Product Generation
C-CMOS	Classical Complementary Metal Oxide Semiconductor
CMOS	Complementary Metal Oxide Semiconductor
CPL	Complementary Pass-Transistor Logic
CPU	Central Processing Unit
CRBBE	Covalent Redundant Binary Booth Encoding
CSA	Carry Save Adder
DPL	Dual Pass Transistor Logic
DSM	Deep Sub-Micron
DSP	Digital Signal Processor
FA	Full Adder
IC	Integrated Circuit
IO	Input/Output
LSI	Large Scale Integration
MAC	Multiply-Accumulator
MOS	Metal Oxide Semiconductor
MOSFET	Metal Oxide Semiconductor Field-Effect Transistor
MUX	Multiplexer
NB	Normal Binary
NBE-RBPPG	Normal Binary Booth Encoding and Redundant Binary Partial Product Generation
NMOS	N-type Metal Oxide Semiconductor
PDA	Personal Digital Assistant
PDP	Power-Delay Product
PMOS	P-type Metal Oxide Semiconductor
PPG	Partial Product Generator
QoS	Quality of Silicon

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

RB	Redundant Binary
RBA	Redundant Binary Adder
RBSD	Redundant Binary Signed Digit
RCA	Ripple Carry Adder
RFIC	Radio Frequency Integrated Circuit
RISC	Reduced Instruction Set Computer
SIMD	Single Instruction Multiple Data
SoC	System-on-Chip
TFA	Transmission Function Adder
TGA	Transmission Gate Adder
ULSI	Ultra-Large Scale Integration
VLSI	Very Large Scale Integration
XOR	Exclusive Or
XNOR	Exclusive Nor

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Chapter 1

Introduction

1.1 Motivation

The Metal Oxide Semiconductor Field-Effect Transistor (MOSFET) was first proposed in 1930's [YEO02]. Integrated circuits (ICs) were found to be much smaller and more power efficient than the discrete components used to build electronics system in 1950's. Around 1980, the Metal Oxide Semiconductor (MOS) technology based integrated circuits became dominant in the market share compared with other technologies, such as bipolar technology. Until 1990, about two thirds of the IC industry sales were of the MOS technology, where more than 90% of them were based on Complementary MOS (CMOS) technology. Nowadays, the technology of IC contains a whole host of innovative devices and systems and performs a wide variety of tasks, whether visible or hidden, which greatly changes the way we live.

In 1960's Gordon Moore, an industry pioneer, presciently presented the now well known Moore's Law [FER89, WOL98], which predicted that the number of transistors in a single integrated circuit chip would double every 18 months. The prediction worked well in the following several decades. To date, the transistor count per chip doubles about once a year [WOL98]. The engineering profession prefers to scale the integrated technology by its orders of magnitude according to the integrated transistor count. Initially, when the circuits contained a few hundred transistors on a single chip, they were called "integrated circuits". As the technology grows, a single chip can achieve an integration of thousands of transistors, which is known as large scale integration (LSI). When the chip grows beyond 10,000 transistors, the term very large scale integration

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

(VLSI) is used. The ultra-large scale integration (ULSI) era advents as a single chip can easily integrates 1,000,000 to 5,000,000 transistors together [FER89]. It is reported in 2000 that a single chip multiprocessor DSP contained 1.6 billion transistors [ACK00]. It is estimated that at the end of 2010's, by using 50nm transistor operating below one volt, a system-on-chip (SoC) will grow to 4 billion transistors [BEN02].

The conventional low-power circuit designers only set their focus on a few niche applications such as wrist watches, pocket calculators, electronic thermometers and some integrated sensors. Currently, the low-power design paradigm is expanding to the area of high performance applications because power dissipation tends to become one of the most important single design constraints. Regardless of how different the reasons the designers lower the power consumption for the targeted applications, minimizing the overall system power has emerged as the number one daunting task. As a matter of fact, the reasons can be zoomed in on two: portability and reliability [ELR97]. Many portable systems that withdraw power from the batteries are used daily, such as laptop computers, digital video cameras, cellular phones, etc. The requirement for low power stems from the needs of both reducing the weight and volume of the devices and lengthening the time to recharge. Reliability is another major concern for high-speed digital systems, which often generate a large amount of heat due to the high clock frequency and denser device integration on chips. Therefore expensive cooling systems are required to ensure the proper functionality of the system. The goal of low power design in this perspective is to lower the system cost and make it operate with higher stability.

With the rapid development of VLSI/ULSI technology where the minimum feature size of the current mature process is shrinking to well below 1μm, or deep sub-micron (DSM) process, the cross-coupling capacitance between adjacent lines of the same layer and that of different layers are becoming more and more important. It has increased from 40% to 70% of the total capacitance when the technology migrates from 0.35μm to 70nm. Besides, the interconnections also increasingly affect the chip power dissipation and create more potential cross talk problem [KHA01, FER89, ELR97].

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

1.2 Objective

The preliminary goal of this research is to investigate and characterize the high performance low power arithmetic circuits and macrocells, which are often the timing closure and power consumption stumbling blocks in digital systems. The techniques for low power design are explored from the device level such as the basic XOR, XNOR gates to the algorithmic level involving digital multiplication and scalar product computation for both the classical two's complement number system and the unaccustomed redundant binary system. Apart from the CMOS circuit level optimization strategies with different logic styles, the development of new circuit structures and the derivation of ingenious architectures from novel algorithms, this research project also aims at providing a new insight into design tradeoffs that involve power, delay and area metrics and establishing a decision model that aids the designer in constructing an arithmetic circuit from its fabrics of favorable characteristics. It is envisaged that the proposed design methodologies, optimization methods and simulation environments are useful in advancing the domain knowledge in VLSI design to the DSM regime.

In a nutshell, the objective of research is to design new arithmetic circuits with the following Quality of Silicon (QoS):

a. low power and high energy efficiency (i.e., low power-delay product)
b. low voltage operation
c. high performance
d. high area efficiency

A complete layout of a sufficiently complex full custom arithmetic macrocell, which is designed based on new algorithm and composed of optimized novel primitive cells, is also targeted on 0.18μm CMOS technology so that more convincing postlayout simulations with back annotated parasitics can be used to validate our proposed design methodologies.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

1.3 Research originality

The major contributions of the research results reported in this thesis can be categorized into the following three major areas: (1) circuit structure (2) algorithm and architecture, and (3) optimization, simulation and testing methods.

In the contributions towards new circuit structure, we have proposed a pass transistor logic design style gate used to efficiently generate the XOR and XNOR functions simultaneously and another gate in a novel complementary CMOS style circuit with regular layout to generate good drivability carry output. Using these two basic gates, we designed two novel 1-bit adder cells, Hybrid-1 and Hybrid-2. The simulation results demonstrate that the improved Module 1, i.e. the proposed XOR-XNOR gate, of Hybrid-2 cell features the most power-efficient cell among a number of current-art 1-bit adder cells over a wide range of supply voltages. The energy efficiency of the Hybrid-2 cell is most pronounced at sub-1V operation. The layout of the Hybrid-2 cell shows that it is also area efficient. As the main building blocks of multipliers, the structures of 4-2 and 5-2 compressors are analyzed and different CMOS logic style circuit implementations of their constituent modules are explored. A novel 5-2 compressor architecture of 4Δ delay is also proposed. The simulation results show that the 4-2 and 5-2 compressors constructed with the novel XOR* cell is able to function down to 0.6V, and features high speed and low power characteristics. Our proposed 5-2 compressor architecture outperforms all the other reported architectures over the range of voltages simulated, particularly when it is configured with the proposed circuits for the XOR* and the carry generator modules. Better performances against other architectures are also attained almost irrespective of the logic styles used for the circuit implementation of their constituent modules.

In the contributions towards new algorithm and architecture, we have formulated a new algorithm for the design of a VLSI circuit for scalar product evaluation. The algorithm leads to a novel full bit-parallel architecture of scalar product macrocell featuring low interconnect complexity, improved power efficiency and highly efficient VLSI area utilization. More importantly, the layout regularity and scalability enhance its

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

performance superiority in deep submicron regime well above conventional VLSI design of vector processing unit for scalar product computation. The overall performances of average power consumption, worst case delay and power efficiency of our post-layout circuit surplus even the pre-layout circuit of the conventional architecture when these circuits are simulated using Synopsys Nanosim over supply voltages from 0.7V to 3.3V based on Chartered CSM 0.18μm CMOS technology. The design consists of 226560 transistors with the core occupying an area of 1430μm×1560μm. The full chip with peripheral circuits occupies an area of 2600μm ×2900μm. The postlayout simulation results of the IP core at 1.8V shows a worst case delay of 6.92ns and a power consumption of 65mW at 50 MHz data rate.

By exploiting the redundant binary (RB) system, another new redundant binary multiplier is proposed based on the newly developed Covalent Redundant Binary Booth Encoding (CRBBE) algorithm. This algorithm fully exploits the characteristics of the Booth encoded numbers to overcome the problem of generating hard multiples and achieves a compatible reduction of RB partial products without inducing any correction vector. Compared with the Redundant Binary Signed Digit (RBSD) multiplier, which is also used to address the hard multiple problem, the proposed CRBBE multiplier has only half the number of partial products to be summed if the same radix Booth encoders are used. Conversely, for the same number of RB partial products, the CRBBE encoder, say Booth-2 CRBBE, and its corresponding RB partial product generator (PPG) are much simpler than that of the RBSD multiplier, say Booth-4 RBSD encoder and RB PPG, which uses many high fan-in gates. The simulation results show that the CRBBE multiplier consumes lower power and computes faster than the normal binary Booth encoder, RB partial product generator (NBE-RBPPG) and RBSD multipliers.

In the contributions towards optimization, simulation and testing methods, we have proposed a systematic optimization procedure to size the transistor of all arithmetic cells in order to perform a fair comparison because the transistor sizing for optimal performance is technology dependent. The optimization procedure employs a greedy

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

sweeping strategy to search for the optimized transistor sizes in iterations, which has been proven to be a convergent algorithm.

During the simulation of full adder, a reasonably simple architecture is proposed to simulate the adder cell in an environment realistic to its actual deployment in the most frequently used parallel multiplier structure. This has overcome the bias in overestimating the performance of the full adders in low voltage operation which are often evaluated in isolation without concern on how they are deployed in actual circuit. Similarly, in order to realistically assess and compare the figures of merits of different configurations of 4-2 and 5-2 compressors at various supply voltages, new simulation environments for the compressors are established to ensure that the measured performances are still sustainable when these cells are integrated in an accumulation tree.

To test the scalar product macrocell, some auxiliary circuits including a unique delay detection circuit are proposed to enable accurate delay measurement of the core with the constraint of limited IO pins. Otherwise, if traditional testing method is employed, many more additional IO pins are required, which is impractical in our case.

The above contributions have led to the publications listed in the author's publications towards the end of the thesis.

1.4 Organization

Seven chapters are presented in the thesis. Chapter 1 exposits the motivation and objectives of our research. The results presented in this thesis are summarized into three categories of significant contributions.

Chapter 2 studies the causes of power consumption in CMOS integrated circuits and the related formulae, followed by a discussion on various methods to achieve low power consumption while sustaining the performance. The characteristic of different logic styles are also presented. This is followed by a detailed discussion on the features and the

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

shortcomings of the various circuit implementations of XOR and XNOR gates, which are the primitive elements used for the construction of full adders and compressors. A study of the different full adder circuits and a brief introduction to the use of high input compressors are provided, which serve as a preface to the more in-depth discussion of Chapters 3 and 4. The last part of this chapter outlines the history and evolution of normal binary and redundant binary digital multipliers, which are fundamentals to the research topics presented in Chapters 5 and 6.

Chapter 3 focuses on the design of full adder cells. It explores the existing full adder designs in different logic styles. Different circuit implementations of the three constituent modules of full adder are analyzed, before the proposed hybrid full adder cells are described. To optimize and compare the performance of the different full adders, a tree structured setup is proposed as the simulation environment. Also, a new transistor optimization procedure is described and the optimized sizing parameters at two different supply voltages for various adder cells being evaluated are provided. The circuits are simulated for power, delay and power-delay product performances and the results are analyzed and compared.

Chapter 4 investigates the performances of several fast 4-2 and 5-2 compressors. The 4-2 compressors constructed around the proposed XOR-XNOR cell exhibit superior power efficiency when they are compared with other configurations of the same architecture. Next, the 5-2 compressor architectures and their underlining building modules are scrutinized. A new fast 5-2 compressor architecture is proposed and its various configurations from different logic styles are investigated and compared against several well known 5-2 compressor architectures. Special setups for the simulation of 4-2 and 5-2 compressors are also described. These setups emulate a realistic application environment that truly reflects its actual operability in a tree-structured multiplier. Finally, the simulation results for various configurations of the proposed architectures and their contenders are presented and analyzed.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Chapter 5 proposes the new algorithm for scalar product computation. Based on the algorithm, the architecture, the floor planning and the delay estimation of scalar product IP core are detailed. The same model of estimation is also applied to the conventional architecture for comparison. The layout in compliance with the CSM 0.18μm CMOS process technology of the proposed scalar product IP core equivalent to 16 full-width multiply-accumulation operations of 16-bit operands and the design of its auxiliary circuit are presented. A comparison of the pre- and post- layout simulation results is provided at the end of the chapter.

Chapter 6 presents our research on the novel redundant binary multiplier. It introduces the notion of RB systems before the structure of the RB multiplier is described in terms of its constituent modules. The existing algorithms and implementations of Booth encoders and partial product generators are explored. An ingenious covalent redundant binary Booth encoding algorithm is proposed. The circuit implementations for different radix encoders and accompanying partial product generators are also described. A 54×54-bit multiplier is constructed based on the proposed covalent redundant binary Booth encoder and RB partial product generator, together with the existing redundant binary adder (RBA) circuit for the RBA summing tree and RB-NB converter. The structure of the RB multiplier using the proposed algorithm is compared with the classic structures of other RB multipliers. Eventually, the simulation results are analyzed and discussed.

Finally, Chapter 7 highlights the virtues of the design methodologies used in the novel low power arithmetic cells and macrocells to conclude the thesis. The potential relevant topics for further research, design challenges and the role of the techniques discussed in the dissertation on advanced technologies are outlined.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Chapter 2

Literature Review

Since our project aims at the design of low power, high performance arithmetic circuits and their applications, a literature review on low power techniques and fundamentals of computer arithmetic circuits is inevitable. This chapter is organized as follows: Section 2.1 studies the causes of power consumption in CMOS integrated circuits and the related formulae, followed by a discussion on different methods to achieve low power consumption while sustaining the performance. The characteristic of different logic styles are also presented. Section 2.2 gives a detailed discussion of the features and the shortcomings of the various circuit implementations of XOR and XNOR gates, which are the primitive elements for full adder. A study of the different full adder circuits and a brief introduction to the use of high input compressors are provided in Section 2.3, which will serve as a preface to the more in depth discussions of Chapters 3 and 4. Section 2.4 narrates the history and evolution of normal binary and redundant binary digital multipliers, which are fundamentals for the research presented in Chapters 5 and 6.

2.1 Low power techniques for digital circuit design

The semiconductor industry has witnessed an explosive growth of the integration of sophisticated personalized devices and multimedia-based applications into mobile electronics gadgetry since the last decade, such as notebook computers, cellular phones, digital cameras, digital video recorders, MP3 music player and personal digital assistants (PDAs). The improved functionality and increased versatility of these portable applications set a stringent requirement on the throughput and a tight constraint on the amount of power that can be dissipated due to the limited battery lifetime. Although new

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

high energy capacity types of rechargeable battery has been developed perpetually, the advancement of the battery technique is far slower than the escalating density of integrated circuits, where millions and millions of transistors are integrated into one chip, driving the semiconductor industry from VLSI era to ULSI era. As CMOS process technology shrinks, the unity gain cut-off frequency, f_T of the transistors becomes comparable with that of the GaAs bipolar technology that it is now practical to design sub-1V radio frequency integrated circuits (RFICs) based solely on the matured low cost, low power CMOS process [TSA02]. The impact of this is that the front end wireless communication circuitries, traditionally based on analog circuit techniques, are now being transferred into the digital domain, which increases by leaps and bounds the rate of data to be processed by the arithmetic circuits in the digital signal processor.

Low power techniques are also important for high-performance processing systems, which may not be portable or battery powered, such as workstations, desktop personal computers, mainframe computers and other applications. Since they rely on ultra high clock speed and extreme chip integration density to deliver the level of performance to remain competitive, the amount of heat generated by the immense computing power is becoming daunting. Lowering the power consumption can reduce the cost associated with packaging, cooling and fans, which are costly to build, operate and maintain for complex cooling requirement. Another incentive for low-power design is on the issue of reliability. There is a well-defined relationship between the junction temperature, the performance and the reliability of an integrated circuit. For example, the delay increases by 14% when the junction temperature increases from 85°C to 125°C [GEO01]. The reliability of an integrated circuit degrades as an exponential function of junction temperature. Silicon interconnect fatigue, package related failure, electrical parameter shift, electromigration, junction fatigue are all sorts of failure mechanism that can be induced by high on-chip temperature and are difficult to manage [BEL95]. Therefore, reducing the power consumption plays a key role in enhancing integrated circuit reliability.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

2.1.1 Overview of power consumption of CMOS circuit

The average power dissipation in digital CMOS circuits is composed of three main sources, which are the dynamic power or switching power consumption, the short-circuit power consumption, and the leakage power consumption, shown in (2.1). In some logic styles, there are continuous current paths between the power supply and the ground, such as bias circuits to provide the necessary operating point to other transistors. In this case, a fourth source, namely, static power is consumed. Since the circuits presented in this thesis use only those logic styles that have no static power dissipation, we will consider only the former three sources of power consumption [BEL95, ELR97].

$$P_{avg} = P_{dynamic} + P_{short\text{-}circuit} + P_{leakage} \tag{2.1}$$

Dynamic or switching power consumption represents the power dissipated during a switching event. In CMOS logic circuits, the dynamic power consumption arises when the voltage source and ground that are connected to the capacitive loads, C_{load1} and C_{load2}, respectively, are charged and discharged through PMOS and NMOS transistors to finish an output voltage transition from logic "0" to "1" and "1" to "0", where logic "1" and "0" are often the supply "vdd" and system ground "gnd". The load C_{load1} exists between the output and PMOS substrates, such as the capacitance between the PMOS drain area and substrate of the driving stage, the capacitance between PMOS gate and substrate of the loading stage, and the capacitance between the output interconnecting lines and power supply lines. Similar situation exists for the load C_{load2}. Fig. 2.1 shows the capacitive load model of an inverter for the calculation of dynamic power consumption.

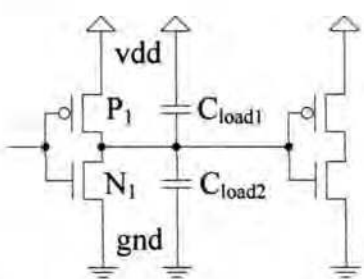


Figure 2.1 *Capacitive load model*

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

For C_{load2}, when the output goes from logic "0" to "1", the power supply charges through PMOS transistor P_1. Part of the energy is dissipated on PMOS and part of it is stored in C_{load2}. At the same time, the already charged C_{load1} is discharged through P_1, releasing the energy previously stored in it. When the output returns from logic "1" to "0", the charged C_{load2} will discharge through the NMOS transistor N_1, dissipating its stored energy. The C_{load1} will be charged again through N_1. Similarly, a part of the energy is stored in C_{load1} waiting to be released in the next "0" to "1" transition, and the other part is dissipated in N_1 during C_{load1} charging. Therefore in each complete cycle of transitions from "0" to "1" to "0", the dissipated power is:

$$C_{load1}V_{dd}V + C_{load2}V_{dd}V = C_{load}V_{dd}V$$

where $C_{load} = C_{load1} + C_{load2}$, and V is the voltage range of the partial transition for the general case.

In general, the number of transitions per unit time corresponds to the system clock frequency f_{clk}, but the node does not transit in all clock cycles. Therefore we define a transition probability, α, that in each clock cycle a node will transit from "0" to "1" and then back to "0". The dynamic power $P_{dynamic\text{-}i}$ for the node i is given by

$$P_{dynamic\text{-}i} = \alpha_i C_{load\text{-}i} V_{dd} V_i f_{clk}$$

For a complete circuit, the dynamic power $P_{dynamic}$ is:

$$P_{dynamic} = \left(\sum_{i=1}^{\#\ of\ nodes} \alpha_i C_{load\text{-}i} V_i \right) V_{dd} f_{clk} \qquad (2.2)$$

The short-circuit power consumption is caused when both the NMOS and the PMOS transistors in the circuit conduct simultaneously for a short amount of time during switching, forming a direct current path between the power supply and the ground. The short-circuit power is inevitable because the real input voltage waveforms always have finite rise and fall time. The short-circuit current, which passes through both the NMOS and PMOS devices during switching, does not contribute to the charging or discharging

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

of the capacitances in the output node. The short-circuit power will be prevalent if the output load capacitance is small, and/or if the input signal rise and fall time is large.

Simply considering a symmetrical CMOS inverter with $k = k_n = k_p$ and $V_T = V_{T,n} = |V_{T,p}|$ for both transistors, and with a very small capacitive load. If the inverter is driven with an input voltage waveform with equal rise and fall time ($\tau = t_{rise} = t_{fall}$), we have the following time-averaged short circuit current provided by the power supply:

$$I_{\text{short-circuit}} = \frac{1}{12} \cdot \frac{k\tau f_{clk}}{V_{dd}} (V_{dd} - 2V_T)^3 \tag{2.3a}$$

Therefore, the short-circuit power dissipation is:

$$P_{\text{short-circuit}} = \frac{1}{12} \cdot k\tau f_{clk} (V_{dd} - 2V_T)^3 \tag{2.3b}$$

It is noted from (2.3b), which is for small capacitive load model, that the short-circuit power dissipation is linearly proportional to the input signal rise and fall time, as well as to the transconductance of the transistors. Hence, to reduce the short-circuit current, an effective way is to shorten the input transition time.

For an inverter with a large capacitive load, the output node voltage will retain its voltage before the input node finishes its transition. Although the currently turned on transistor has large voltage difference between its drain and source, the power dissipated is now contributed to the dynamic component because the other transistor has been turned off.

There are two types of leakage currents: reverse-bias diode leakage on the transistor drains and sub-threshold leakage through the channel of an off device. For the first type, the diode leakage occurs when one transistor is turned off but another transistor pulls up or down the drain of the former transistor, making it reversed biased with respect to the bulk potential. For example, when an inverter has a logic "1" input, the NMOS transistor is turned on pulling the drains of both PMOS and NMOS to low voltage. At same time,

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

the PMOS transistor is turned off with its drain-to-bulk voltage equal to $-V_{dd}$. The resulting current is

$$I_{reverse} = A_D J_S \left(e^{\frac{qV_{bias}}{kT}} - 1 \right), \tag{2.4}$$

where V_{bias} is the reverse bias voltage magnitude on the PN junction, A_D is the area of the drain diffusion, and J_S is the reverse saturation current density, which is dependent on the technology and weakly dependent on the supply voltage. Usually the reverse-bias diode leakage current is a small fraction of the total power consumption in most chips. But it may be significant for a chip spending most of its time in standby operation, since this power is always being dissipated even when there is no switching activity.

The second type of the leakage power is the subthreshold leakage which occurs due to carrier diffusion between the source and the drain when the gate-source voltage, V_{gs}, is between the weak inversion point and the threshold voltage V_t , where carrier drift is dominant. The current in the subthreshold region is given by:

$$I_{ds(subthreshold)} = Ke^{\frac{V_{gs}-V_t}{nV_T}} \left(1 - e^{\frac{V_{ds}}{V_T}} \right) \tag{2.5}$$

where K is a function of the technology, V_T is the thermal voltage (KT/q) and V_t is the threshold voltage. For $V_{ds} >> V_T$, $(1-e^{-V_{ds}/V_T}) \approx 1$; that is, the drain to source leakage current is independent of the drain-source voltage V_{ds}, for V_{ds} slightly larger than 0.1V.

2.1.2 Low power design techniques

Several methods can be applied to optimize the power dissipation of digital systems. These methods can be divided into five different levels, which are device or process level, circuit or logic level, architecture level, algorithm level, and system level. Device characteristics, device geometries and interconnect properties are the significant factors

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

in lowering the power consumption. Circuit-level measures such as the proper choice of circuit design styles, reduction of the voltage swing and clocking strategies can be used to reduce power dissipation at the transistor level. Architecture-level measures include smart power management of various system blocks, utilization of pipelining and parallelism, and design of bus structures. Fig. 2.2 shows the hierarchical levels of optimization techniques.

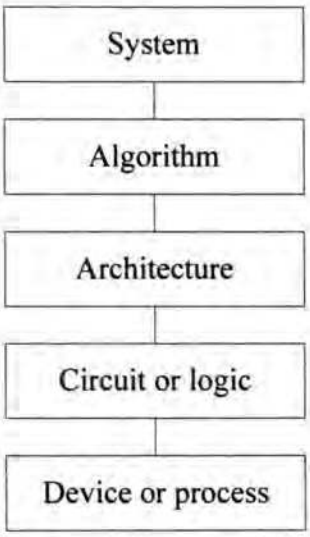


Figure 2.2 *Power optimization hierarchy*

The switching power dissipation in CMOS digital integrated circuits is a strong function of the power supply voltage, approximately quadratic to V_{dd}. Therefore, reduction of V_{dd} is a lucid way to effectively reduce the power consumption. However, the savings in power dissipation comes with a significant increase in circuit delay. For a CMOS inverter, the propagation delay is given by:

$$\tau = \frac{C_{load}}{k_n (V_{dd} - V_{T,n})} \left[\frac{2V_{T,n}}{V_{dd} - V_{T,n}} + \ln\left(\frac{4(V_{dd} - V_{T,n})}{V_{dd}} - 1 \right) \right] \tag{2.6}$$

The delay τ is increased when the power supply voltage V_{dd} is decreased provided all other parameters remain constant. To compensate for the negative effect of lowering supply voltage on the circuit performance, it is a feasible means at device level to scale

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

down the threshold voltage of the transistor accordingly. However, at some point, the threshold voltage and supply voltage reduction is offset by an increase in the leakage currents, resulting in the existence of an optimal threshold voltage for a given level of logic complexity. In so doing, the effect of reducing noise margin must also be taken into consideration.

The circuit level optimization techniques involve optimizing place and route, transistor sizing, reduced swing logic, logic minimization, logic level power down and etc. As semiconductor technology advances, the effect of the parasitic capacitance of the interconnects on the latency, power consumption and VLSI area becomes non-trivial in submicron and deep submicron technologies [KAT00, SYL98]. As IC designs move into the system-on-chip era, the continuous quest for more functionality and increasing emphasis on dedicated computational-intensive intellectual property cores has resulted in aggressive downscaling of the transistors to provide higher levels of integration. This has the positive effect of reducing the internal gate delay, but at the same time, the parasitic capacitance of the interconnect wires increases commensurately. It is observed that, for a 0.35μm technology, the gate delay is about 100ps, local interconnection delay is about 150ps, and global interconnection is about 1000ps. If the technology advances to 0.07μm, the gate delay is decreased to 10ps, local interconnection delay is decreased to 50ps, whereas the global interconnection is increased to 6000ps. The coupling capacitance between adjacent lines separated by minimum spacing increases from 40% to 70% [KAT00]. Sylvester and Kuetzer [SYL98] reported that, for a 1mm metal-1 line, the RC delay in a 0.5μm technology was 15ps as compared to 340ps in a 0.1μm technology. In high performance, power-hungry VLSI and ULSI digital circuits, reducing the parasitic capacitance has emerged as a key design premise for the reduction of dynamic power dissipation, which accounts for more than 90% of the total power in a CMOS device. The parasitic capacitance can be reduced by using fewer and smaller devices as well as fewer and shorter interconnects. Glitches and spurious transitions can also be minimized by equalizing the delay paths to the gate inputs [NAN99]. Therefore, at the layout level, the place and route should be optimized such that signals that have high switching activity

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

(such as clocks) should be assigned short wires and signals with lower switching activity are allowed to use relatively longer wires.

Also in circuit level, optimized transistor sizing plays a vital role in reducing power consumption. To achieve low power, it is important to equalize all delay paths so that a single critical path does not unnecessarily limit the performance of the entire circuit. As voltage varies, the optimal sizing of the circuits for low power operation is quite different from that for high speed applications.

The logic style helps to optimize the circuit in low power operations [BLA94, CHA92]. Many CMOS structures have been used to implement logic functions. Dynamic logic circuits and static logic circuits are two main categories of the CMOS logic circuits [VAI01, MUR96, QUI01, PIG95, ZIM97]. The static CMOS logic circuits are further divided into complementary CMOS and pass-transistor logic circuits.

Dynamic CMOS circuits are clocked and operated in two phases, a precharge phase and an evaluation phase. Despite being capable of fast evaluation with reduced number of transistors compared to its static CMOS counterpart, its large clock loads and high switching activities due to the precharge mechanism make dynamic CMOS circuits consume more power [ZIM97]. When both the precharge PMOS and discharge NMOS logic circuits are in the off state, the dynamic circuit has floating nodes which are vulnerable to alpha particle attacks. The operation margin of a dynamic circuit narrows rapidly with reduction of supply voltage due to its noise susceptibility. Security devices are usually included to guide against noise corrupting the data on the floating nodes, but their inclusion causes considerable delay time penalty [MUR96].

Complementary (conventional) CMOS circuits are made of a PMOS pull-up network and a NMOS pull-down network controlled by the same set of inputs feeding to the gates of the PMOS/NMOS pairs [VAI01, ZIM97]. The series transistors in the output stage form a weak output driver. This could be corrected by additional output buffers/inverters. The advantage of the Complementary CMOS is its robustness against voltage scaling and

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

transistor sizing, which are essential to provide reliable operation at low voltage and arbitrary transistor sizes. Moreover, layout of the Complementary CMOS circuit is straightforward and area-efficient due to the complementary transistor pairs and smaller number of interconnecting wires.

The basic difference between the pass-transistor logic [ASS91] and the Complementary CMOS logic styles is that the source side of the pass logic transistor network is connected to some input signals instead of the power lines [VAI01, ZIM97]. The advantage is that one pass-transistor network (either PMOS or NMOS) is sufficient to implement the logic function, which results in smaller number of transistors and smaller input load. However, pass-transistor logic has an inherent threshold voltage drop problem. The output is a weak logic "1" when "1" is passed through a NMOS and is a weak logic "0" when "0" is passed through a PMOS. Unlike Complementary CMOS, pass-transistor logic is sensitive to voltage scaling and transistor sizing, thus limiting its robustness. In other words, efficiency and reliable operation of logic gates are not necessarily guaranteed at low voltage or reduced transistors' sizes [RAB96]. Thus, transistor sizing is crucial for correct gate operation and the optimization process is less intuitive. Usually, the layout of pass-transistor logic circuit is not as straightforward, as it requires more space to segregate the different diffusion areas, which increases the routing complexity of the interconnecting lines. Consequently, it has poor area efficiency due to irregular transistor arrangements and complex wirings.

Transmission gate logic circuit is a special kind of pass-transistor logic circuit [VAI01, WES93]. It is built by connecting a PMOS transistor and a NMOS transistor in parallel, which are controlled by complementary control signals. Both the PMOS and NMOS transistors will provide the path to the input logic "1" or "0", respectively when they are turned on simultaneously. Thus, there is no voltage drop problem whether the 1 or 0 is passed through it. The main disadvantage of transmission gate logic is that it requires double the number of transistors or more transistors than the standard pass-transistor logic to implement the same circuit. Chapters 3 and 4 of this thesis will further

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

demonstrate how the circuits of basic gates and arithmetic cells are optimized in the circuit/logic level.

There is a great deal of freedom in optimizing architectures for low-power including power shutting down of the unused blocks, using parallel processing blocks and pipelining techniques. For example, by duplicating the operating unit, the speed requirement is decreased by a factor of 2, which means the power supply voltage can be lowered while maintaining the required throughput. The total capacitance is increased by a factor of about 2. The lowered power dissipation is mainly brought by the lowering of the supply voltage. Similar situation applies on the pipelined circuits. But the two obvious consequences of this approach are the greater area and the higher latency.

In the algorithmic approach, power consumption is effectively minimized by decreasing the number of operations to decrease the hardware usage and the required operational speed. A good algorithm also makes the hardware design straightforward and regular.

In the system level optimization, the clocks used system wide can be set to a low frequency, while the on-chip phase locked loop provides the circuit with the required high frequency clock. High level integration of off-chip peripheral integrated circuits into one single chip is also a viable alternative, such as the System-on-Chip (SOC).

Chapters 5 and 6 will demonstrate, by means of practical arithmetic applications, the low power optimization approaches at architectural and algorithmic levels.

2.2 Design of XOR and XNOR cells

When the power supply voltage is lowered to decrease the power dissipation, various problems due to the low voltage arise, such as driving capability weakening, logic failure due to output voltage drop in some logic styles, noise margin reduction and so on. Therefore, the design of the basic gates needs to be carefully tailored to ensure that they will be functioning in low voltage low power applications.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

The exclusive or (XOR) and exclusive nor (XNOR) gates are the most frequently used fundamental units in a variety of digital circuits, such as half adder, full adder, comparator, parity checker, filters, multipliers and so forth. The functions of XOR and XNOR gates are shown in (2.7) and (2.8) respectively:

$$A \oplus B = A\overline{B} + \overline{A}B \tag{2.7}$$

$$A \odot B = AB + \overline{A}\,\overline{B} \tag{2.8}$$

Many circuits of XOR and XNOR are proposed in the last decade [BUI00(1), BUI00(2), BUI02, CHE99, FAN96, LEE97(2), RAD01, VES99(1), VES99(2), WAN94, YU00]. Fig. 2.3 shows the classical CMOS design of XOR circuit, which implements the Boolean function (2.9). This design uses many transistors, occupies comparatively more silicon area and the delay is longer.

$$\begin{aligned} A \oplus B &= \overline{A\overline{B}}\left(A + B\right) \\ &= \left(\overline{A} + \overline{B}\right)\left(A + B\right) = A\overline{B} + \overline{A}B \end{aligned} \tag{2.9}$$

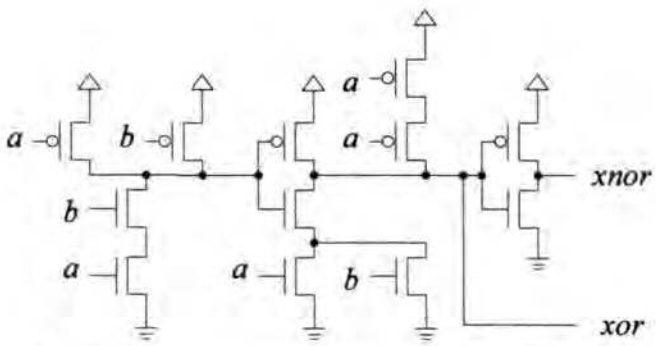


Figure 2.3 *Classical CMOS XOR-XNOR gate*

The mirror CMOS styled circuits are shown in Fig. 2.4. It is simpler than the classical CMOS version, but it needs complementary signals for all its inputs, which implies two additional inverters are necessary to generate them.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

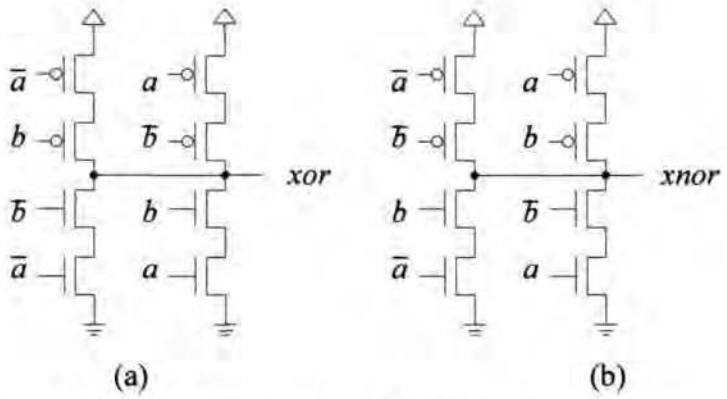


Figure 2.4 *Mirror CMOS XOR and XNOR gates*

Wang et. al. [WAN94] proposed conceivably the most frequently used XOR gate and XNOR gate designs. The goal was to use non-complementary inputs to generate the needed functions. These designs are so important and fundamental that many other designs of XOR and XNOR gates are improved and optimized based on them. The cross back PMOS transistors provide a strong "1" output when the inputs are "01" or "10". When input with "11", the two series NMOS transistors pull down the output with strong signal while the cross back PMOS's are turned off. The only exception is when the input is "00", a weak "0" is passed to the output by the PMOS's. A threshold voltage drop problem is encountered when the circuit is used in low voltage applications. The XNOR circuit encounters the same problem with the input of "11". This problem gives room for further improvement by other researchers.

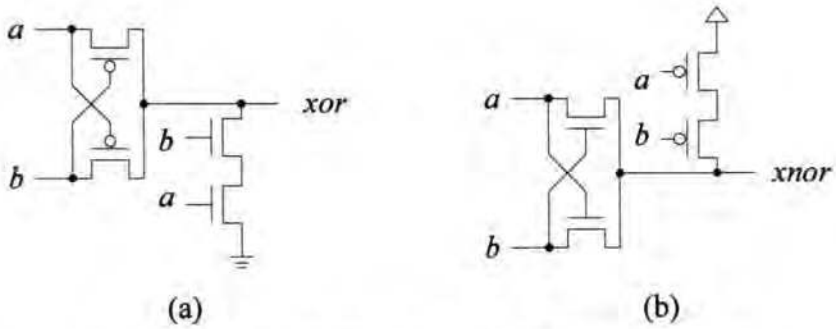


Figure 2.5 *Wang's Cross-back XOR and XNOR gates*

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Fig. 2.6 shows one modified version of Wang's cross back XOR and XNOR gates. One NMOS and one PMOS transistors are added to the previous XOR and XNOR gates, respectively, to address the weak logic problems. When the input "00" is exerted on the XOR gate of Fig. 2.6(a), the added NMOS turns on to pass strong logic "0" to the output. Now they are able to work at very low voltage. The price to solve this problem is the introduction of one additional complementary input signal.

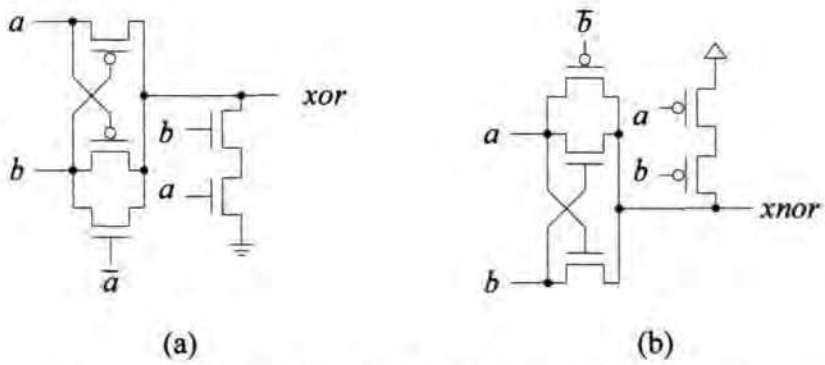


Figure 2.6 *Modified Cross-back XOR and XNOR gates*

Fig. 2.7 shows the CMOS+ XOR and XNOR gates, each of which is composed of one transmission gate and two pass transistors. In the XOR circuit design, when input B is "0", the transmission gate is turned on with the help of the complementary input of B, and input A is passed on to the output to deliver the strong logic. When the input B is "1", the transmission gate is turned off, and the two pass transistors function as an inverter. So the inverted signal of input A appears on the output. In any case, the output always has strong logic.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

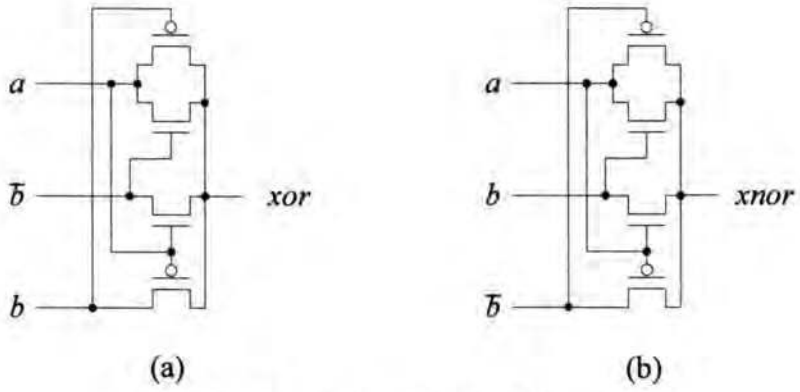


Figure 2.7 *CMOS+ XOR and XNOR gates*

Fig. 2.8 shows a similar but modified design of CMOS+ circuits in Fig. 2.7. The two pass transistors are replaced by a gated inverter, whose control terminals are connected to the complementary inputs.

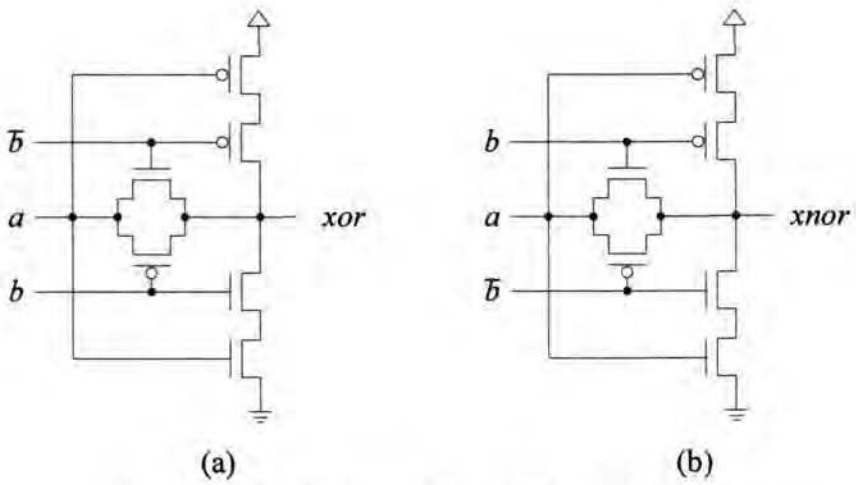


Figure 2.8 *Modified CMOS+ XOR and XNOR gates*

If the XOR and XNOR gates are implemented in CPL logic style, simpler circuits are obtained in Fig. 2.9, where only two transistors are used to realize the functions. Unfortunately, they suffer from the weak logic problems when the output is "1". Nevertheless, they are fundamental to the construction of other enhanced circuits.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

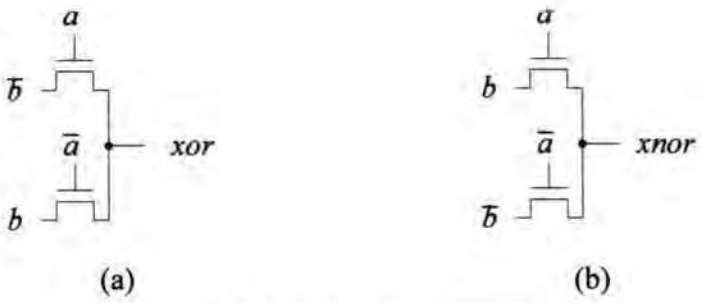


Figure 2.9 *CPL XOR and XNOR gates*

To overcome the non-full swing problem, a weak feedback latch, which consists of one inverter and one PMOS transistor for feedback shown in Fig. 2.10, is inserted between the output and the CPL circuit. With the latch, the output will always provide the strong signals and is suitable for cascading. However, the circuits cannot work at very low voltage because the latch may be driven by a weak logic signal at the beginning of some input transitions. When the supply voltage is sufficiently low, the weak signal does not have enough drivability to turn on the latch.

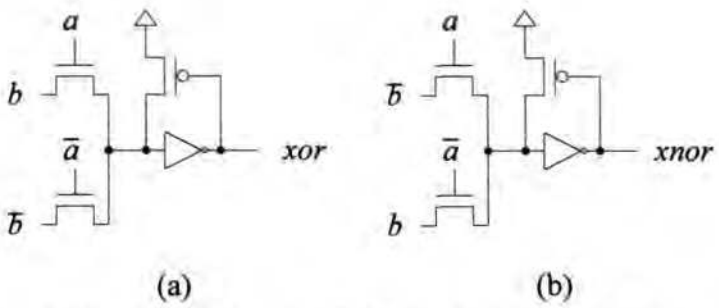


Figure 2.10 *Latched CPL XOR and XNOR gates*

The DPL style XOR and XNOR gates, as shown in Fig. 2.11, eliminates the weak logic by adding the corresponding PMOS counterparts to each NMOS transistors. Although each input needs a pair of complementary signals, the preceding stage that drive the inputs of these circuits experiences balance load because each input signal of these circuits is connected to one transistor's gate and another transistor's source.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

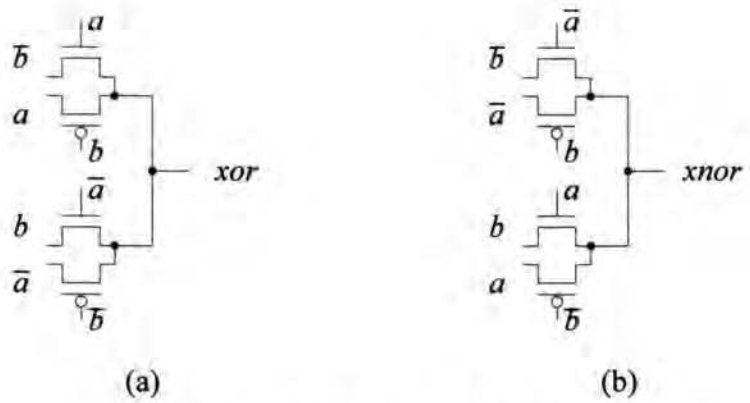


Figure 2.11 *DPL XOR and XNOR gates*

The transmission gate XOR and XNOR, as shown in Fig. 2.12, are very similar to the DPL style, except that their input loads are not as balanced as those of the DPL circuits.

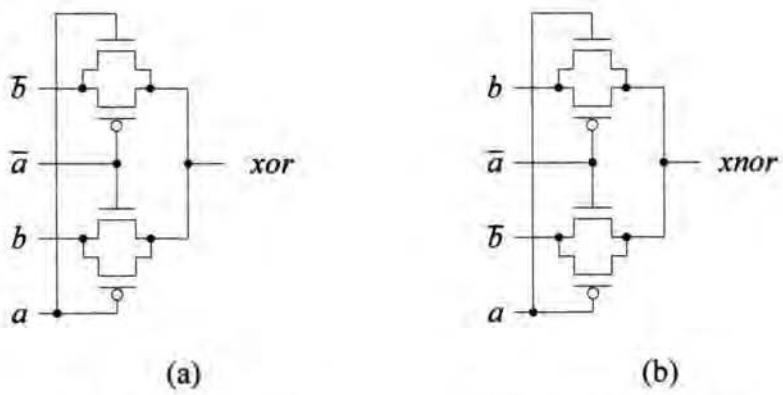
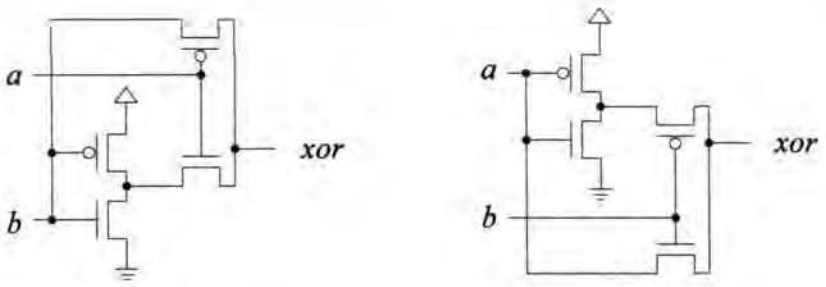


Figure 2.12 *Transmission gate XOR and XNOR gates*

Fig. 2.13 shows yet another pass transistor style circuit. One inverter is used to generate the complementary signal of one input. The two pass transistors function either as another inverter to generate the strong output or as two separate pass transistors with one of them passes the weak signal. Therefore, they are also not suitable for low voltage applications.

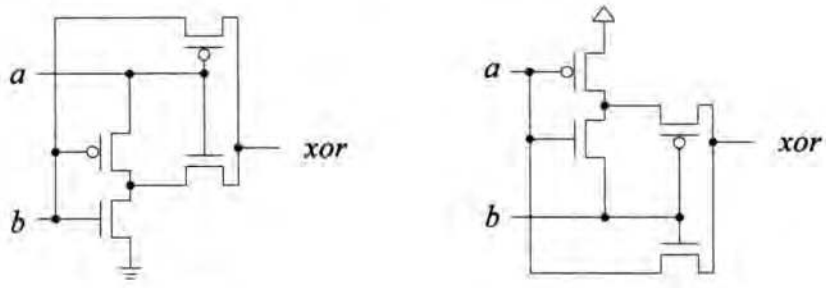
ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library



(a) (b)

Figure 2.13 *Pass transistor XOR and XNOR gates*

A modified pass transistor XOR and XNOR gates, named as powerless-XOR and groundless-XNOR, are shown in Fig. 2.14. The difference between the circuits of Fig. 2.14 and those of Fig. 2.13 is that the original power connected PMOS transistor in XOR and the original ground connected NMOS transistor in XNOR are now connected to one of the input signals. Therefore some power is saved compared with the circuits of Fig. 2.13, but the weak logic problem still exists.



(a) (b)

Figure 2.14 *Powerless-XOR and groundless-XNOR gates*

For brevity, we only show the functional part of each XOR and XNOR designs from Fig. 2.4 to 2.14, without adding inverters or buffers to the outputs. But many of the designs will suffer from inadequate driving capability even if they can output strong logics, such as DPL style, transmission gate style, etc. To improve the drivability, it is a common practice to add inverter in the design, such as inverter connected XNOR to realize XOR with improved driving capability, or vise versa, as shown in Fig. 2.15. The circuits with weak logic outputs can also be enhanced to some extent by adding output inverters.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

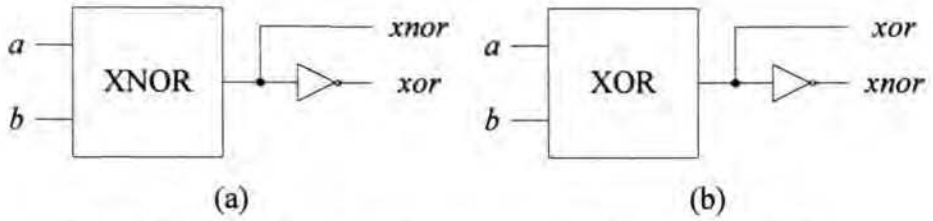


Figure 2.15 *Drivability enhancements for XOR and XNOR gates*

In some circuits, it is not enough to have independent XOR or XNOR gate with single output, while the complementary outputs of both XOR and XNOR are preferred. The methods in Fig. 2.15 using inverters to generate complementary outputs also fit these requirements. But, the outputs are not balanced in terms of timing and/or driving capability. Therefore, several dedicated XOR-XNOR gates are designed. One of them is the modified CPL style, which is shown in Fig. 2.16. The previously presented CPL XOR and XNOR gates are connected together by a pair of feedback PMOS transistors, which restores both outputs to strong signals. This is because for any input combination, at least one output is strong and drives the other output through the feedback PMOS transistors to give the strong signal. The drawback is that complementary pairs of inputs are still needed.

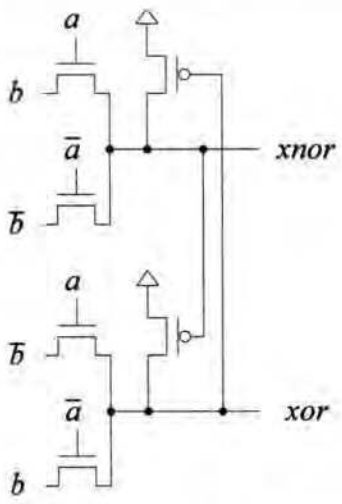


Figure 2.16 *Modified CPL XOR-XNOR gate*

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Another solution is to combine Wang's XOR and XNOR circuits together and logically optimize it to produce the circuit shown in Fig. 2.17. However, this design does not perform well at low voltage, especially during the transition from "01" to "00" or from "10" to "11". A detailed analysis is presented in Chapter 3.

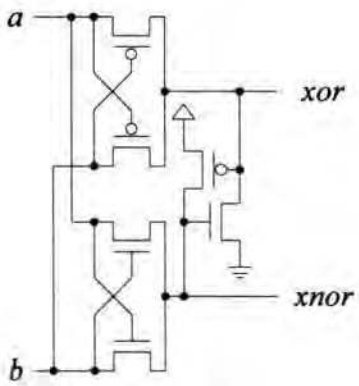


Figure 2.17 *Feedback XOR-XNOR gate*

One solution of the problem is to use Cheng's XOR-XNOR gate [CHE99], where two pass transistors and an inverter are added. When the input transits from "01" to "00" or from "10" to "11", one of the added pass transistors will turn on and a strong logic (i.e., strong "0" through NMOS or strong "1" through PMOS) is passed to drive the feedback transistors.

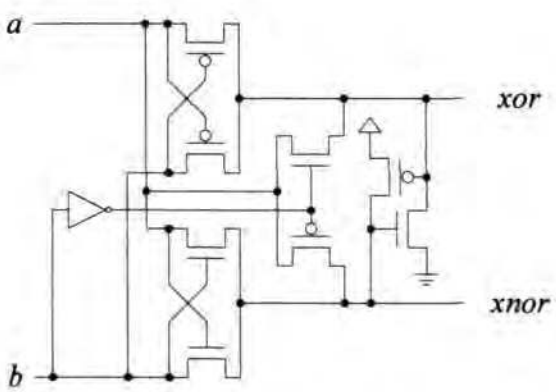


Figure 2.18 *Cheng's XOR-XNOR gate*

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Since a full adder will add three bits, the sum output of the result is an XOR of these input bits or an XNOR of them, as shown in (2.10)

$$sum = a \oplus b \oplus c = a \odot b \odot c \tag{2.10}$$

Some papers proposed 3-input XOR gates for the sum generation. Several designs are indeed the cascade of the XOR or XNOR gates presented earlier. There are two dedicated designs of 3-input XOR gates. One is the mirror CMOS style, which is shown in Fig. 2.19. Similar to the XOR gate, every input requires a complementary signal pair. The design uses 20 transistors without counting the number of transistors for the inverters.

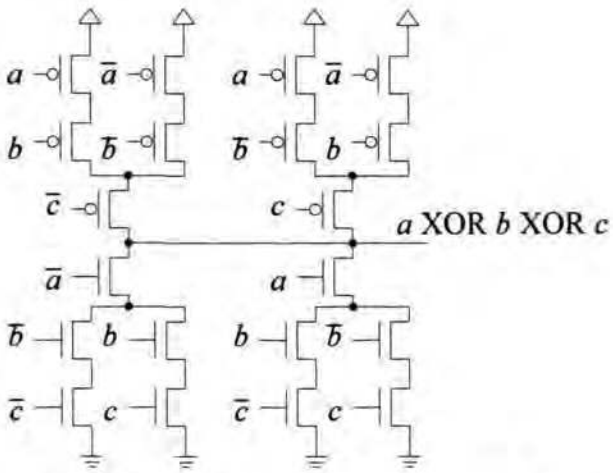


Figure 2.19 *Mirror CMOS 3-input XOR gate*

Another design is proposed by Fang [FAN96]. Two pairs of PMOS cross back transistors and two pairs of NMOS cross back transistors are used. However, the circuit has weak logic problem on both its internal nodes and the output nodes. It fails to work at low voltage, especially when the three inputs are "000" or "111".

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

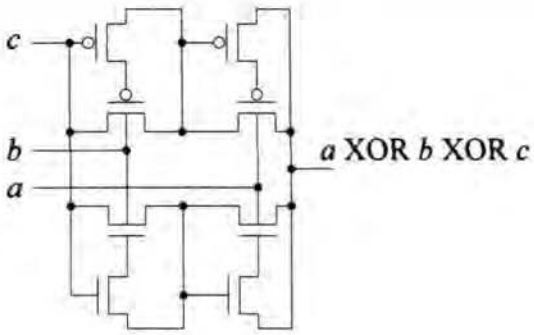


Figure 2.20 *Fang's 3-input XOR gate*

Due to the unsatisfactory performances of the existing circuits of 3-input XOR gate, they are not popular in arithmetic circuit design.

2.3 Design of full adders

One bit full adder is another fundamental arithmetic unit widely used for addition, multiplication, digital filtering operations in microprocessors and digital signal processors. From its function equations, full adder is composed of exclusive-or circuits and carry generation circuits. A variety of different designs of full adders have been reported [BUI00(2), BUI02, CHE99, FAN96, KO95, LU01, QUI01, RAD01, RAD99, SAY02, SHA97, SHA98, SHA99, SHA00, SHA02, VES99(1), WAN94, WEY02, YU00, ZHA03, ZHU92, ZIM97].

Fig. 2.21 shows the classical CMOS style full adder. It uses two parts of the complex gates to generate the carry and sum signals. Each part is the implementation of the basic Boolean equations of the full adder function, where the sum generation utilizes the logic of the carry generation circuit.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

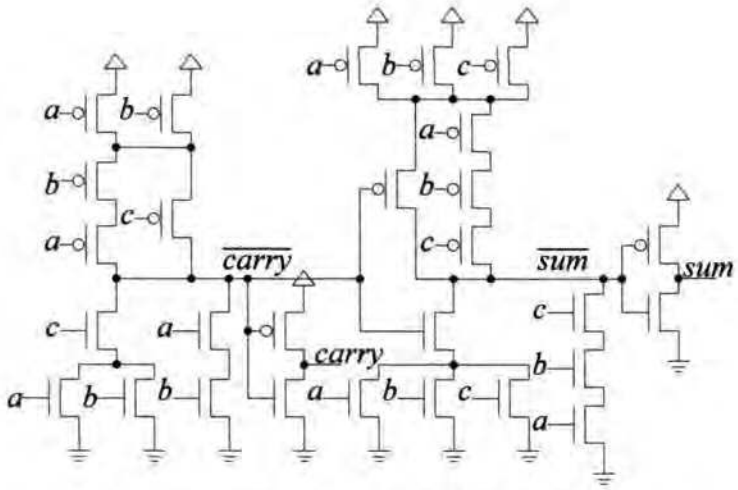


Figure 2.21 *Classical CMOS full adder*

Fig. 2.22 is the mirror CMOS circuit, which is an optimized version of the classical CMOS circuit. The PMOS transistors block mirrors the NMOS transistors block. Both the classical and mirror CMOS full adder circuit are of 28 transistors each and each of their inputs is made of non-differential logic, which is preferable in many applications.

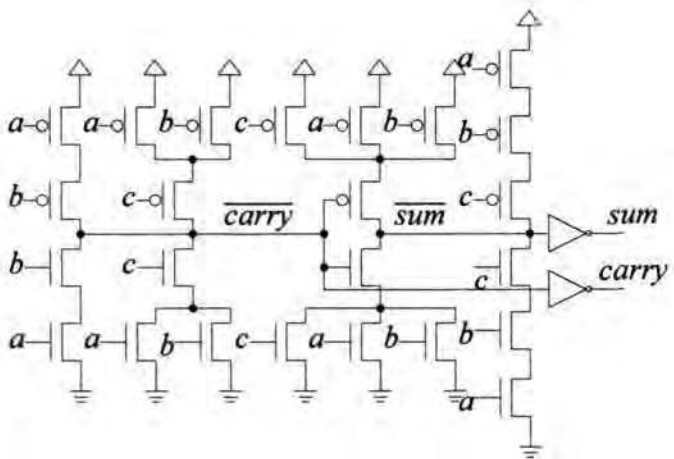


Figure 2.22 *Mirror CMOS full adder*

Fig. 2.23 shows a CMOS full adder with fast carry, where there is one XOR gate delay from the carry input to the sum output and two NAND gates delay from the carry input to the carry output.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

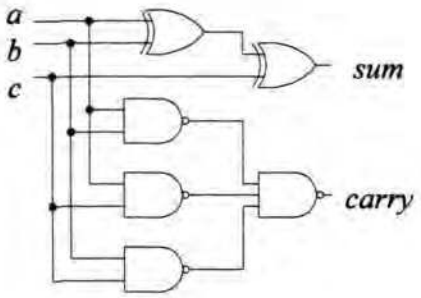


Figure 2.23 *Fast carry CMOS full adder*

A modified CPL style full adder is shown in Fig. 2.24, which uses two independent circuits with a total of 32 transistors to realize the generation of sum and carry. Since all input and output signals are differential, the modified CPL full adders can be cascaded, such as in the tree structure multiplier, without the need of any additional inverters to generate the complementary signals.

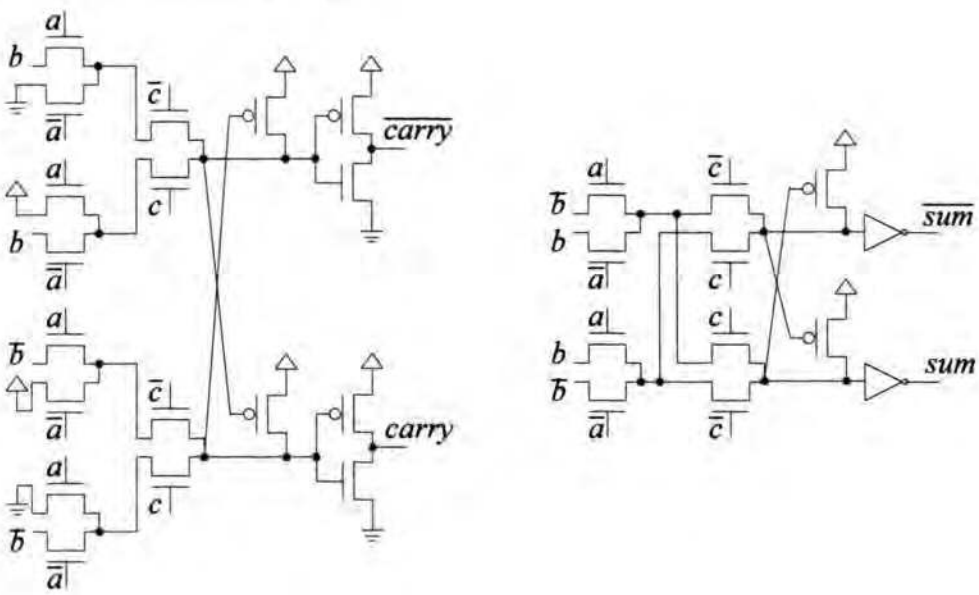


Figure 2.24 *Modified CPL full adder*

Fig. 2.25 shows the 1-bit full adder of latched CPL style using 22 transistors, which is a direct latched CPL translation of the modified CPL full adder. As opposed to the modified CPL style, the input of latched CPL full adder is differential while the output is non-differential. Besides, although the outputs are strengthened by the weak feedback

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

latches, several internal nodes of the latched CPL style full adder still suffer from the weak logic problem.

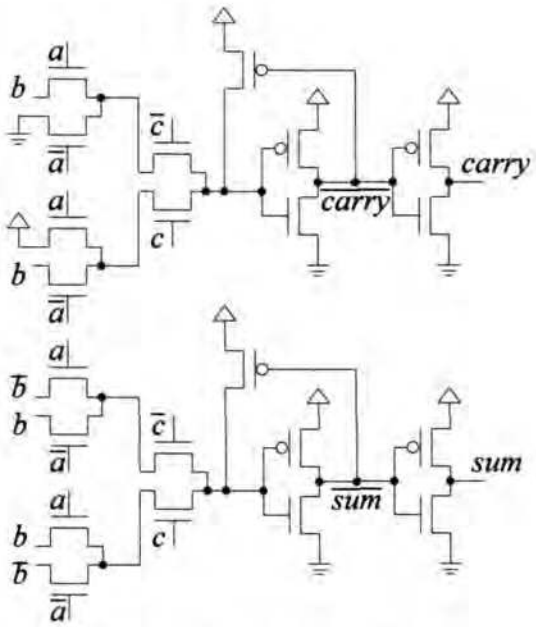


Figure 2.25 *Latched CPL full adder*

A large amount of recently published full adders have a three-module structure shown in Fig. 2.26, which is analyzed by Shams and Bayoumi [SHA97]. Module 1 in the structure generates the internal complementary signals, which are the exclusive-or and exclusive-nor of two of the inputs, usually inputs *a* and *b*. Based on the two internal signals, the design of the sum generator in Module 2 and the carry generator in Module 3 are simplified and optimized. Therefore, many authors proposed new circuits for each of these modules and combined them together to form new full adders. Some typical cases are discussed in Chapter 3.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

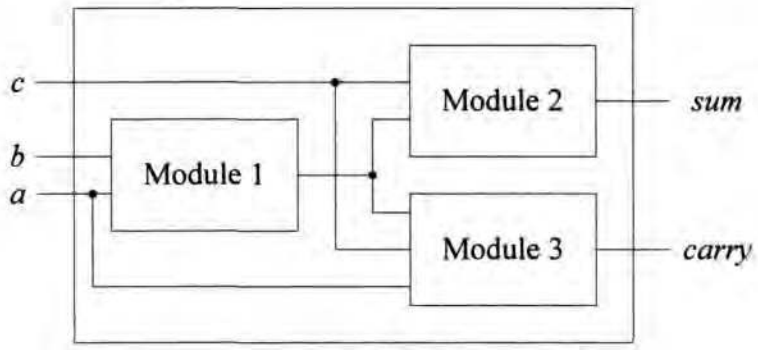


Figure 2.26 *Structure of full adder*

In multipliers, full adders are used as 3-2 counters to compress the partial products in the carry save adder tree. In order to achieve more regular structure and lower latency of the partial product accumulation stage in the multiplier, 4-2 and 5-2 compressors have been widely employed nowadays for high speed multipliers. Owing to its regular interconnection, 4-2 compressor is ideal for the construction of regularly structured Wallace tree with low complexity [RAD00, WAN95]. Several 4-2 compressor circuits have been proposed for low power applications [GU03(1), HSI98, MAR99, MEH91, PRA01, RAD00]. Some of them are able to operate at low supply voltages but require excessive number of transistors due to their complementary CMOS structures, others use smaller number of transistors but fail to function at ultra low voltage, or lack the driving capability to drive the next level of subcircuits. Higher input compressors have also been studied by researchers [GU03(2), KWO00, MEH91, PRA01] and fast 5-2 compressors have been increasingly employed in large word-size multipliers and high precision multiply-accumulators [HSI98, KWO00, RAD00]. Analogous to the analysis of full adders, the 4-2 and 5-2 compressors can also be decomposed into several modules [GU03(2), KWO00, PRA01]. The detailed structures and performance will be studied in Chapter 4.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

2.4 Digital multiplier architectures

2.4.1 Normal binary multipliers

Multipliers are widely used in many areas. Digital filters and multimedia signal processing applications often involve a large mount of multiplications. In different types of microprocessors, multiplier is an indispensable core in the arithmetic logic unit (ALU) and floating point co-processors. Two basic operations are involved in digital multiplier, which are the generation of partial products and their accumulation. There are two ways to speed up multiplication. One is to reduce the number of partial products; the other is to accelerate the accumulation. Obviously, a smaller number of partial products also reduce the complexity and computation time to accumulate the partial products. The multiplication can be considered as a series of repeated addition. So, multipliers are more expensive, operate relatively more slowly and consume more power than adder or subtractor.

The multiplier architectures can be broadly classified into two categories: serial and parallel, as shown in Fig. 2.27. The chip area, computational speed and power dissipation are critical factors for multiplier design [ABU96]. As both operands are input to the multiplier serially, the circuitry of a serial multiplier [PAR00, PEK03, LU95, CHA97] is small. Hence, the chip area, the hardware cost as well as the power consumption can be minimized. However, serial multiplier is much slower than parallel multipliers. Pipelining is a way to increase the speed of serial multiplier [CHO02, FAN00].

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

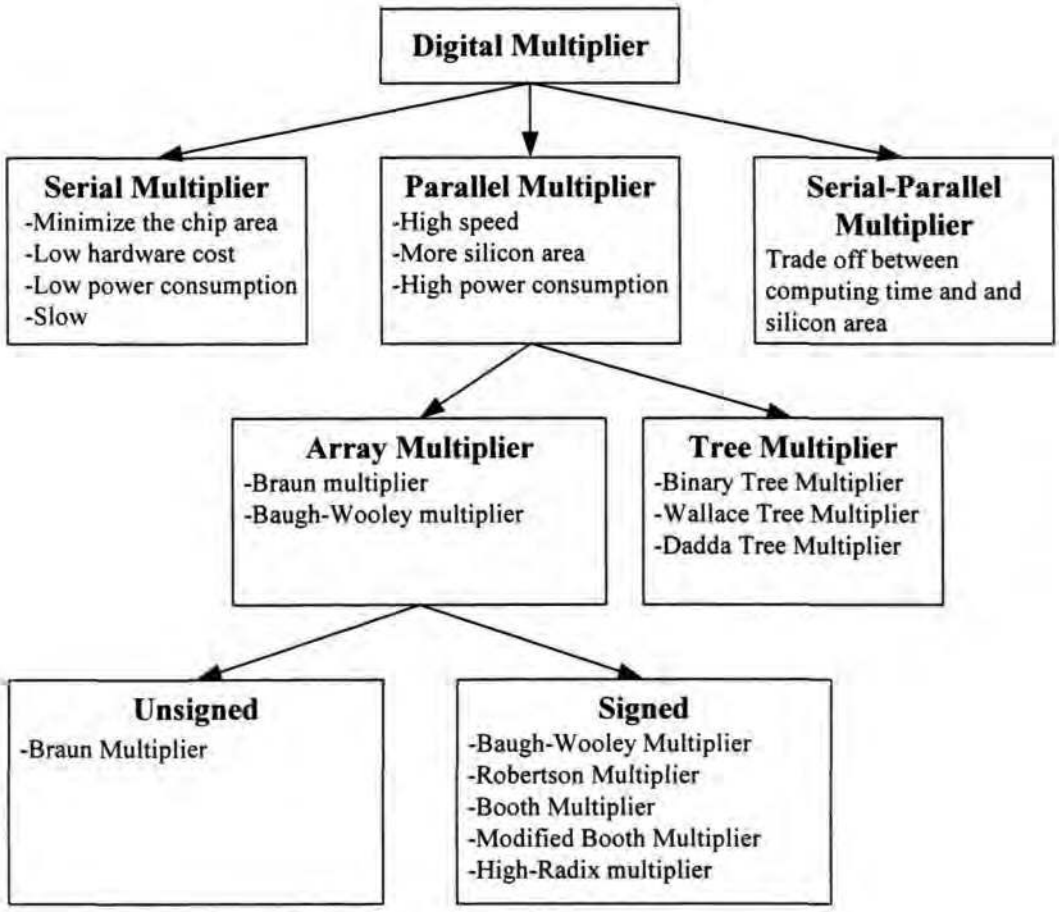


Figure 2.27 *Classification of multipliers*

In parallel multipliers [MEH91, WAL64, WAN95], both operands are input to the multiplier in a parallel manner. The circuitry uses larger area and is more complex than the serial multipliers. There are two types of parallel multipliers: array multiplier [PAR00, RAB01, BRA63, BAU73] and tree multiplier [DAD65, WAL64]. Array multipliers such as Braun multiplier [BRA63] and Baugh Wooley multiplier [BAU73] have regular layout, whereas tree multipliers such as Dadda [DAD65] and Wallace [WAL64] multiplier have higher speed. These multipliers are widely used in reduced instruction set computer (RISC) CPUs, DSPs and graphics accelerators.

Another hybrid type of multiplier is called serial-parallel multiplier [PAR00], which is used to trade between hardware simplification and computing speed. In this architecture,

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

one operand is entered serially and the other is stored in parallel with a fixed number of bits.

The Braun multiplier, invented by Braun Edward Louis in 1963 [BRA63], is a relatively simple form of parallel multipliers. It is a direct implementation of the paper-and-pencil method as how we would perform the multiplication by hand. Braun multiplier is also commonly known as the carry save array (CSA) multiplier. This multiplier is well suited for multiplying two unsigned numbers. The iterative structure consists of an array of AND gates and adders without any sequential logic or registers. The regular layout makes it ideal for VLSI and ASIC realization.

The Baugh-Wooley Multiplier was designed by Bruce A. Wooley and Charles R. Baugh in 1973 [BAU73]. This multiplier is actually an improved version of the Braun Multiplier as the hardware structures of both multipliers are very similar. However, Baugh-Wooley multiplier is able to operate with both the unsigned and signed numbers. It is conjectured that the invention of Baugh-Wooley multiplier has contributed significantly to the advent of computer arithmetic, because it is the first fast multiplier capable of performing both unsigned and signed multiplications. Although Baugh-Wooley multiplier is time consuming and less efficient when dealing with large operands, it is nonetheless a good candidature even as of today when the operands are less than 16 bits.

The first tree multiplier was introduced in 1964 by Wallace [WAL64]. He suggested a notion of a carry-save adder (CSA) tree as a way to efficiently and progressively reduce the multi–operand additions in the multiplication process to a final stage of two operand addition. The Wallace tree multiplier employs full and half adders to add up the partial products simultaneously in a parallel sequence. Later, Dadda [DAD65] suggested an optimal compression scheme using different size counters (mainly 3-2 counters, which are full adders and 2-2 counters, which are half adders) and showed that different schemes of cell allocation, including the one introduced by Wallace require different number of cells (counters). A natural VLSI layout of either Wallace or Dadda architecture is to distribute the cells such that the lengths of the interconnections are as

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

short as possible. However, as the sum and carry signals need to be communicated to non-adjacent cells and propagate downwards across non-adjacent stages, the wiring and layout are irregular and more complicated than the array multipliers, and the long wiring interconnections have the potential of introducing crosstalk and markedly reduce the performance under DSM process technology [LEE97(1), KAT00, SYL98, WAL00].

As early as 1951, A. D. Booth [BOO51] introduced the Booth multiplier. The Booth algorithm provides a simple way to generate the product of two signed binary numbers by means of the Radix-2 arithmetic. The drawback of Booth's algorithm is that it becomes inefficient when there is a great number of isolated 1's in the operands. In 1961, Macsoreley [MAC61] proposed the modified Booth algorithm, which is also known as Radix-4 Booth encoding. Since the modified Booth algorithm is capable of reducing the number of partial products by half, it makes efficient hardware implementation of digital multipliers with reduced logic depth and logic complexity. Since its introduction, the modified Booth algorithm has soon evolved to become a ubiquitous algorithm in prevailing high-speed multipliers, especially for those that have to operate with large operands.

Most hardware architectures take advantage of the two basic operations in the multiplication algorithm, i.e., the generation of the partial products and the accumulation of these partial products. Thus, there are two lucid ways to speed up multiplication. One is to reduce the number of partial products [OKL96]; the other is to accelerate the accumulation process by minimizing its latency [PAR97, SRI92]. High radix Booth algorithm has the advantage of reducing the number of partial products to be added, while the Wallace/Dadda approach speeds up the addition of the partial products. When these two techniques are combined in a hybrid fashion, they can yield a multiplier that is much faster than the traditional Wallace/Dadda or Booth multiplier [MIL92, KHO99]. Today, this method is commonly used to realize high-speed multiplier because theoretically, it is the fastest solution. However, due to the complicated inter and intra stage interconnections, the VLSI layout of the adder tree has been irregular and obscured by the mingling cell connectivity, leading to highly inefficient utilization of silicon area.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Furthermore, as the size of the multiplier increases, the interconnection problem becomes exponentially complicated and difficult to optimize.

To avoid the cross-stage interconnection, one method is to use the 4-2 compressors [GU03(1), MAR99, RAD00, WEI81] instead of the conventional full adders (also known as 3-2 counters) to reduce the partial products. This approach is first introduced by Weinberger [WEI81] in 1981 as a means to speedup the column compressions of the dot matrix representation of the adder tree in parallel multipliers since a 4-2 compressor can reduce four inputs of the same weight to two. It produces a much more regular structure than the design that based on the 3-2 counter. This has the effect of drastically simplifying the interconnection because the partial products are added up in the form of a binary tree without having the wires leap frog across cells of non-adjacent stages. However, the avoidance of cross-stage interconnection by 4-2 compressors is achieved at the expense of long lateral communication wiring within each stage. Song and Micheli developed the 6-2 and 9-2 high input compressor families, which are built on 4-2 compressors and 3-2 counters [SON91]. Although the speed of multiplication can be in principle increased by employing higher order counters or compressors to reduce the number of adder tree stages, there is an optimal number of partial products (depending on the operand width and the radix of Booth encoding used) below which the complexity and latency of the high order counter and compressor cells offset the benefits they provided [BON96, GU03(2), PRA01, SON91]. It should be noted that the height and width of the silicon area, on first order estimation, depends on the number of stages and the width of the stage with the largest number of cells. High input compressors are wider and larger than the simple 3-2 counter, the maximum length of wiring for multipliers built on high input compressors may exceed the width ratios for optimal area efficiency [WAN95].

2.4.2 Redundant binary multipliers

To seek for other alternatives of carry propagation free accumulation, the redundant binary (RB) representation, which is one of the signed digit representations, was firstly

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

introduced by Avizienis [AVI61] in 1961. Takagi [TAK85] proposed to apply this new arithmetic for fast multiplication and Edamatsu [EDA88] implemented it in VLSI. For the RB partial product accumulation, its physical design is regular and simpler than that of the Wallace tree multiplier due to its good repeatability and elimination of the cross-stage interconnections of differently weighted digits. In view of the non-trivial effect of the parasitic capacitance of the interconnects on the latency, power consumption, VLSI area and crosstalk problem in deep submicron devices [SYL98, KAT00], there has been a rekindle of interests in RB multipliers [SHI97] owing to the fine granularity of the RBAs and its carry free addition property [HAR87, MAK96, SAK00, BES02(1), BES02(2)].

Unfortunately, there have been a number of new RB multiplier proposals which claimed to have superior performance, were later proved to be incorrect or misleading in the fundamental concept. In the Radix-64 Booth encoded RB multiplier proposed by [LEE02], the authors justified that Radix-64 Booth encoding was the optimum radix because the result of their proposed RB multiplier shows an excellent performance in comparison to other published results based on different radix Booth encoders. However, these comparisons have not been carried out compatibly under the same process technology. It is obvious that in their scheme the critical delay of the Booth Encoding and the Partial Product Generation (BEPPG) stages contribute to almost 41% of the total delay time, which is much higher than the 26% as reported in [MAK96]. Therefore, in terms of the transistor counts, their scheme is the minimum. As for Radix-64, the claim that this is the optimum choice in Booth encoding is definitely not sustainable. In [KIM01] a carry-free equivalent bit conversion algorithm was proposed for the RB-to-NB conversion in an attempt to eliminate the final carry propagate adder. However, due to a flaw in the truth table for the carry-free RB-to-NB conversion algorithm, a carry chain in the conversion stage was erroneously neglected [KIM03]. The errors have been detected by several other researchers [ERC03, RUL03] and it was proven that carry propagation is ineluctable in any multiplication process [RUL03]. As a matter of fact, for most RB multiplier, the critical path includes the RB-to-NB conversion. In [CHO01], a direct-conversion scheme was also proposed without any carry propagation to minimize this critical path for parallel architectures. Despite the latency of this converter is a

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

constant independent of word size, carry propagation have been re-introduced into the revised adding rule. Therefore, the declaration has been misleading as the original carry-free addition property was completely abolished in the RBA tree of this multiplier.

We can safely conclude from the above experience that the parallel transformation from any redundant number representation to binary number without incurring some degree of carry propagation is impossible. Some mechanism has to be catered for the carry ripple. It is a matter of trade-off and efficiency. In RB multiplier, it is preferred to maintain the carry-free addition property in the RBA tree instead of annihilating this property by changing the adding rules to improve the reverse conversion efficiency. With the carry-free RBA tree, the carry propagation will inevitably be imposed on the final RB-to-NB conversion stage to some extent. New RB multiplier architecture, particularly one based on novel RB Booth encoders and partial product generators, is one of our research topics detailed in Chapter 6.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Chapter 3

Design and Performance Evaluation of Low Voltage Low Power Full Adder Cells

3.1 Introduction

Adder is an elementary unit of DSP. It is used in ALU, floating-point unit, digital filters, linear transformations, correlators, etc. Full and half adder cells are the cores dominating the critical paths of complex arithmetic operations like subtraction, multiplication, division, exponentiation, etc [CHE94, KO95, KWO00, NAN99, REN93, SHA02, SON91]. Thus, their performance and complexity at transistor level directly influence the overall performance of the system. Due to the pervasive use of portable electronic devices, and the cost of current packaging technologies, power dissipation has become a vital design criterion. Very often, the optimization tasks are restricted by how best the arithmetic operators are implemented in the cell library provided to the designer for the synthesis. The total power consumption of the system will decrease when power efficient adder cells are used in the critical path of compound arithmetic circuits.

In this chapter, two new low voltage low power full adder cells are proposed. Both cells employ the same complementary CMOS based carry out generation circuit coupled with a chosen sum generation circuit. These fabrics of hybrid logic design styles feature not only the full swing and balanced outputs but also strong output drivability. The increase in area due to higher transistor count of complementary CMOS output stage is largely compensated by its area efficient layout. To enable the adder to operate at ultra-low

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

supply voltage, the pass logic circuit that co-generates the intermediate XOR and XNOR outputs has been improved in the second adder cell to overcome the threshold voltage drop problem. A cascaded circuit structure is introduced to simulate the proposed 1-bit full adder cells and several other recently reported designs operating in a realistic application environment. Simulation results based on the TSMC 0.18μm CMOS process parameters show that some survival cells in stand alone operation may still fail when cascaded in a larger circuit, either due to the lack of drivability or unsatisfactory speed of operation. Our proposed full adder cells are proven to be power-efficient through rigorous simulations.

The role of full adders in computer arithmetic can be classified into two main categories. One category involves the chain structured applications [ALI02, CHA95], such as the ripple carry adders (RCA). In these applications, the critical path often traverses from the carry-in to the carry-out of the full adders. It is demanded that the generation of the carry-out signal be fast. Otherwise, the slower carry-out generation will not only extend the worst case delay, but also create more glitches in the later stages, hence dissipate more power. The other category involves the tree structured applications, which is frequently used in Wallace tree multipliers, array multipliers and multiplierless digital filters [CHA95, SON91, WAN95]. Full adders in these applications form a tree of several layers to compress the partial products to one stored carry format number before a final carry propagation adder converts it to a normal two's complement number. For high speed and low power operation, it is required that the outputs, i.e., the sum and carry-out of the full adder be generated simultaneously to minimize the glitches in the lower stages. The published literatures on full adder pay no attention to the specific requirements and the stringent drivability of the latter application. In this chapter, we target the tree structured application for the evaluation of full adders with the optimization and simulation pursued in the proposed tree structure simulation environment. Two new adder cells composing of hybrid CMOS logic styles in its constituent modules are proposed for use in the applications of the second category. For a fair comparison with other full adder cells, a systematic and unified approach to size the transistors of different full adder cells for optimal power-delay product performance is also suggested.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

This chapter is organized as follows: Section 3.2 explores the existing full adder designs in different logic styles. In Section 3.3, different circuits for three constituent modules of the full adder are analyzed. It is followed by the description of the proposed hybrid full adder cells. To optimize and compare the performance of different full adders, a tree structure setup is proposed for the simulation environment in Section 3.4. Also, a new transistor optimization procedure is described and the optimized sizing parameters at two different supply voltages for various adder cells being evaluated are provided. The circuits are simulated for power, delay and power-delay product performances and the results are analyzed and compared. Section 3.5 summarizes and concludes the work presented in this chapter.

3.2 Review of existing full adder cells

A full adder has three inputs and two outputs shown in Fig. 3.1. The three inputs, *a*, *b*, *c*, and one of the outputs, *sum*, are of the same weight, while the other output *carry* weighs one binary bit higher than the others. Therefore, their relationship is shown in (3.1):

$$a + b + c = sum + 2 \cdot carry \tag{3.1}$$

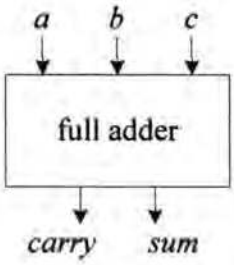


Figure 3.1 *Block diagram of full adder*

The following equations are most frequently utilized to generate the outputs.

$$sum = a \oplus b \oplus c \tag{3.2}$$

$$\begin{aligned} carry &= ab + bc + ac \\ &= ab + (a \oplus b)c \end{aligned} \tag{3.3}$$

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Fig. 3.2 shows the low power 1-bit adder cells [RAD01, SHA99, SHA02, ZIM97] implemented in several variants of static CMOS logic styles. The classical CMOS full adder (C-CMOS) consisting of 28 transistors (henceforth '*n* transistors' is abbreviated as *n*T) is shown in Fig. 3.2(a). It is based on the mirrored CMOS structure with PMOS pull-up and NMOS pull-down transistors. The buffers at the last stage provide the required driving power to the cascaded cells. However, the two outputs are not balanced because $\overline{carry}$ is used to generate the internal signal $\overline{sum}$, causing the output *sum* often generated later than the output *carry*. The skew of the sum and carry outputs causes the carry-save outputs of the Wallace/Dadda tree [WAN95] to have irregular input arrival profile, making it difficult to exploit the advantage of fast carry propagate adder. The delay of a fast carry propagate adder accounts for 25% to 35% of the total tree structured multiplier delay and its power/delay are typically characterized based on even or unimodal input arrival profile. Equalizing the different paths of the adder cells in a tree-structured multiplier can also help to minimize glitches to conserve power.

Fig. 3.2(b) shows the complementary pass transistor logic (CPL) [CHA95] full adder with swing restoration. Its dual rail structure uses 32 transistors. Output inverters are also used to ensure its drivability. A transmission function full adder (TFA) [ZHU92] based on the transmission function theory is shown in Fig 3.2(c). A transmission gate adder (TGA) [WES93] in Fig. 3.2(d) uses CMOS transmission gates to fulfill the XOR and multiplex functions. The two latter circuits have fewer transistors than the previous two full adders. Even lesser transistor count adder circuits have been reported, mostly exploiting the non-full swing pass transistors with swing restored transmission gate techniques. This is exemplified by the state-of-the-art design of 14T [VES99(1)] in Fig. 3.2(e) and 10T [BUI02] in Fig. 3.2(f). The above adders differ not only in their transistor counts but also in the way their intermediate signals are generated. They will be simulated under the same process technology and environment to compare with our proposed 1-bit full adder cells on various key performance factors.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

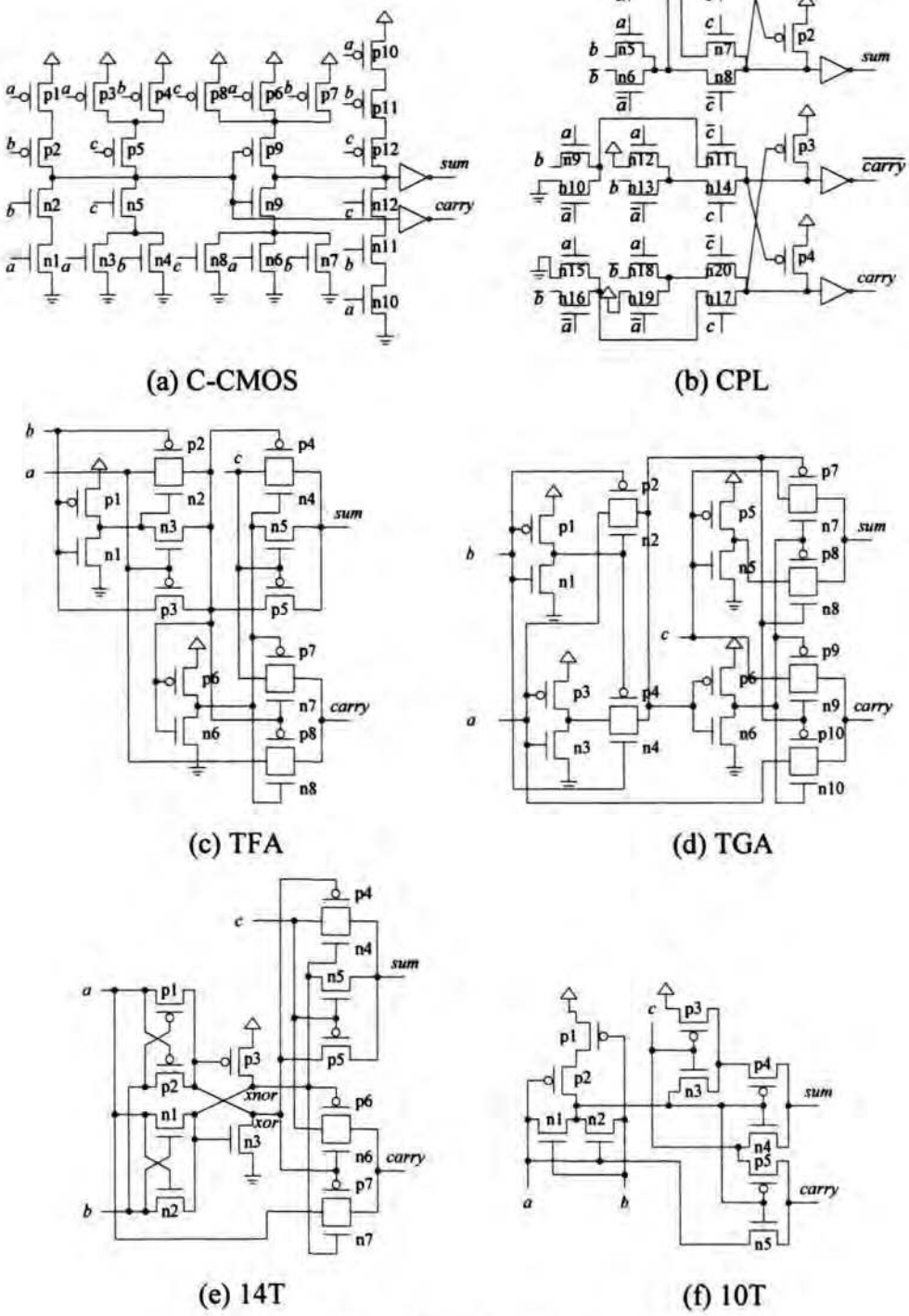


Figure 3.2 *Existing full adder cells*

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Unfortunately, different logic styles tend to favor one performance aspect at the expense of the other. Therefore, besides power, delay and power-delay product, other performance criteria, such as the working voltage range, outputs balancing, full voltage swing at any nodes, and driving capability [BLA94, VAI01, SHA99, SHA02, ZIM97] are taken into consideration in the design and evaluation of adder cells for the tree-structured applications.

3.3 Full adder architecture and its building blocks

Most full adder designs, such as Fig. 3.2(c) to Fig. 3.2(f), can be partitioned into three modules for analysis. They are the XOR/XNOR block to generate the XOR and/or XNOR signals to control the other blocks, the XOR block to generate the *sum* output, and the carry generator block, which is usually implemented in multiplexer, to output the *carry*. The block diagram is shown in Fig. 3.3.

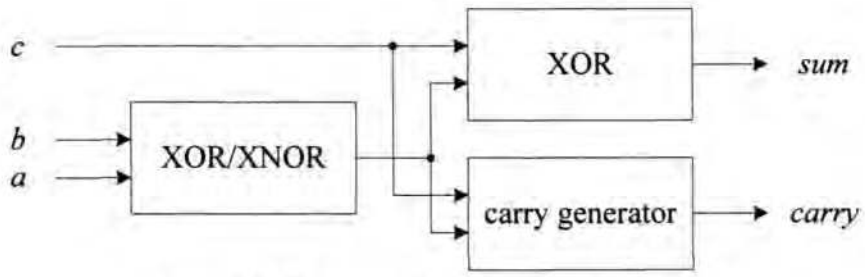


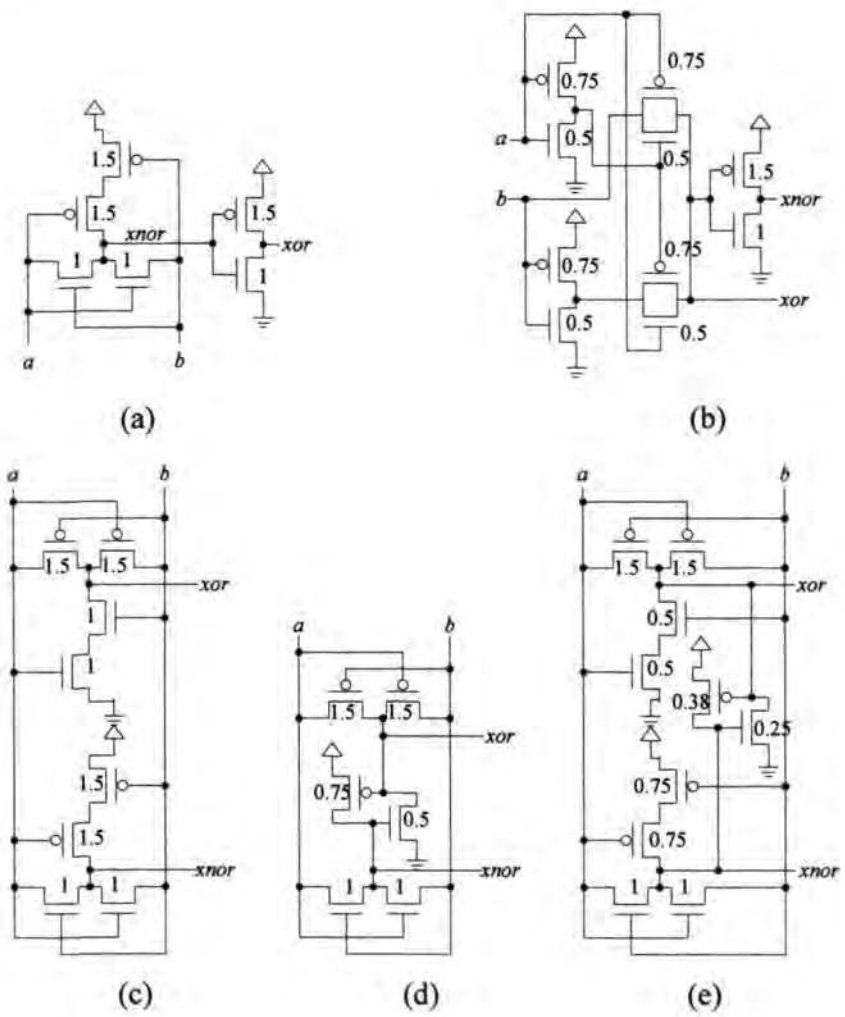
Figure 3.3 *Three-module structured full adder*

A full adder is considered as a 3-2 counter when it is used as the carry-saved adder (CSA) in tree-structured multiplier. Some of the circuits used to implement its constituent modules are also applicable to similar modules of high input counters or compressors used in the CSA tree. The design of 4-2 and 5-2 compressors is the main topic of the next chapter. Relevant circuits presented in this section will also be referred by the next chapter.

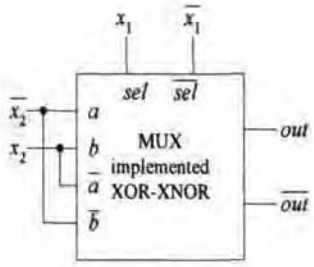
ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

3.3.1 XOR/XNOR module

Fig. 3.4 shows several designs [BUI02, RAD01, RAD00, SHA97, SHA00, SHA02] for the XOR/XNOR modules. The numbers next to the transistors are their optimized sizes used for the compressors described in Chapter 4. The design of Fig. 3.4(a) has the least number of transistors and consumes very low power [SHA97, SHA00, SHA02]. However, it generates a weak logic '1' at the *xnor* node when the primary inputs are both '1's, which prevents it from functioning reliably at low supply voltage. The design of Fig. 3.4(b) is able to operate at low voltage, but it is not power efficient [SHA97, SHA00, SHA02]. Both designs, Fig. 3.4(a) and 3.4(b), use inverters to generate the complementary XOR and XNOR signals, therefore, their outputs skew heavily in time.



ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library



(f)

Figure 3.4 *Circuit implementations of XOR/XNOR module*

The design of Fig. 3.4(c) consists of two cross back XOR and XNOR cells [SHA97, SHA00, SHA02]. It is able to generate the complementary XOR and XNOR outputs simultaneously. However, it performs non full-swing operations for some input patterns causing their corresponding outputs to be degraded by $1V_{th}$. For example, the *xnor* output transmits a weak logic "1" when both inputs are "1"s, whereas the *xor* output transmits a weak logic "0" when both inputs are "0"s. When the power supply voltage is lower than 1V, the week logic transmission will slow down the charging and discharging speed of the driven circuits, or worse, unable to turn on or off the driven transistors as desired. Therefore, it is also not a suitable candidate for low voltage operation.

The combined XOR-XNOR cell of Fig. 3.4(d) was proposed in [RAD01, RAD00]. It is a low power circuit with the least number of transistors that can output XOR and XNOR concurrently. It eliminates the transmission of weak logic for certain input patterns by virtue of the feedback PMOS-NMOS transistors in the midst of the circuit. Nevertheless, it is still not suitable for low voltage applications for the following reason. When the inputs change from "01" or "10" to "00" or "11", the feedback transistors that were turned off originally will be turned on by both a weak logic driver and a high impedance driver. This transition takes a long time at very low voltage slightly above $2V_{thp}$. Let us analyze the process of input changing from "01" or "10" to "00". When the current input is "01" or "10", logic "0" is passed through the NMOS transistor and logic "1" through the PMOS transistor. Both of the feedback transistors are in off status. However, when the next input "00" arrives, a weak "0" ($V = V_{thp}$) is passed through two PMOS pass transistors to the *xor* output, while the *xnor* output stays at high impedance at the

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

beginning of the transitions. This weak "0" turns on the feedback PMOS so that the *xnor* output is pulled up to logic "1", which turns on the feedback NMOS to discharge the *xor* output completely to ground. In the first half of the transition, the feedback transistors are driven by the cross-back transistors, while in the next half, the feedback transistors can drive themselves by positive feedback. Thus a staircase shaped voltage waveform occurs during the transition process in the output node *xor*, as shown in Fig. 3.5. Similar situation happens at the transition from "01" or "10" to "11", too. Owing to the lower V_{thn} and high electron mobility of the NMOS transistor, the entire process is faster than the previous case, which relies heavily on the PMOS switch. Anyway, these slow transitions increase the short circuit current of the following stage tremendously, which lead to a rise in the power dissipation.

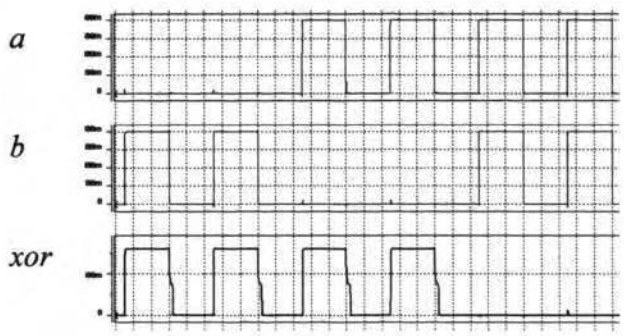


Figure 3.5 *Slow step output transitions when ab change from 01, 10 to 00 for the XOR/XNOR circuit of Fig. 3.4(d) at supply voltage of 0.8V*

To overcome the weakness of the circuit in Fig. 3.4(d), we proposed a new circuit for the XOR/XNOR module. The proposed circuit is shown in Fig. 3.4(e), which is also able to generate the *xor* and *xnor* outputs simultaneously. Two series PMOS are added to solve the slow transition problem of changing from "01" to "00" while another two series NMOS are added to solve the problem of changing from "10" to "11". When the inputs changed to "00", the *xnor* output obtains a strong "1" through two series pull-up PMOS to the power supply, avoiding the high-impedance state happening in Fig. 3.4(d). Similarly, the *xor* output obtains a strong "0" through two series pull-down NMOS to ground when the inputs changed to "11". Therefore, for any input combination, at least one node is of strong logic to drive the feedback transistors. Hence, the proposed circuit

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

is rather robust against delay driven voltage scaling. Fig. 3.4(f) shows the use of a dual-rail multiplexer, such as CPL and DPL multiplexers, to construct the XOR-XNOR module, which can be used for the fully multiplexer-based implementation [PRA01].

3.3.2 XOR module for sum output

The XOR module to generate the *sum* output usually does not need to provide the XNOR output, but it should provide sufficient output current to drive the next stage of full adders or compressors (to be discussed in the next chapter). Fig. 3.6 shows two possible XOR gate designs for this simple XOR module [SHA97, SHA00, SHA02]. Fig. 3.6(a) is a low power design of the XOR function, but the limited drivability prevents it from being used as an output module in full adder and compressor. Fig. 3.6(b) is an XNOR circuit followed by an inverter, which has stronger driving capability than Fig. 3.6(a). The numbers enclosed in brackets are the sizes of the transistors used in the 5-2 compressors described in Chapter 4.

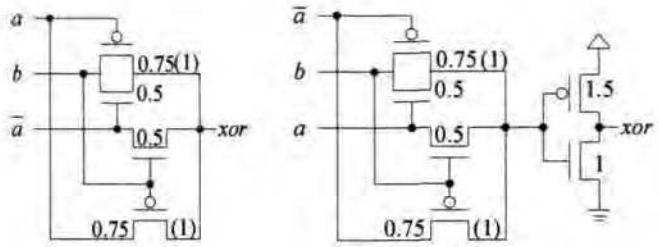


Figure 3.6 *XOR module circuits for sum output*

3.3.3 Carry generator module

The carry generator modules are usually implemented by multiplexers. Several designs are shown in Fig. 3.7. Fig. 3.7(a) is widely used in low power full adder cells [SHA97, SHA00, SHA02]. However, its driving capability is somewhat limited, which causes signal decay when many stages are to be cascaded. So it is not suitable for use in the tree structured application. Fig. 3.7(b) improves the driving capability by adding an output buffer at the expense of increasing its power dissipation [SHA97, SHA00, SHA02]. The output buffer formed by the cascaded inverters is designed such that the first inverter is

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

downsized by halved that of the output inverter in order to cut down the power dissipation.

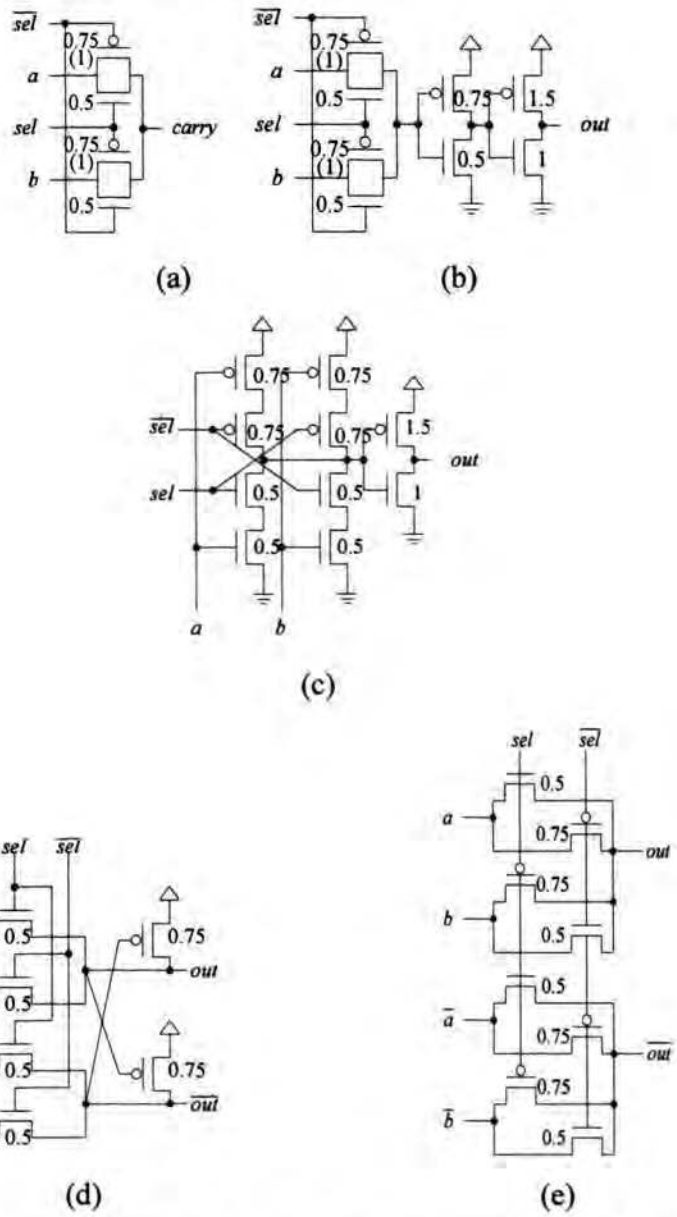


Figure 3.7 *Implementations of the carry generator module using MUX*

The circuit of Fig. 3.7(c) is a multiplexer implemented in standard complementary CMOS logic style [GU03(1), ZIM97]. Being a complementary CMOS circuit, it is robust against both voltage scaling and transistor sizing. Despite having one inverter lesser than

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

the design of Fig. 3.7(b), this circuit still delivers sufficient drive to its succeeding circuits through the output inverter. The total number of transistors of this circuit is 10. Although it is two more than that of Fig. 3.7(b), the silicon areas occupied by both circuits are almost the same. This is because Fig. 3.7(b) requires more space to segregate the different diffusion areas, which increases the routing complexity of the interconnecting lines. The regularity of the layout of the circuit of Fig. 3.7(c) is evident from the diagram on its right. The multiplexer circuits of Fig. 3.7(d) and Fig. 3.7(e) are implemented in CPL and DPL logic styles, respectively [PRA01, ZIM97]. As they are dual-rail circuits, complementary pairs of primary inputs and outputs need to be generated. Although they can generate full-swing outputs, due to the pass transistor structure, they will not provide adequate drivability if many such circuits are cascaded, particularly at low supply voltage. Therefore if these circuits are used at the output ports of the full adder or compressor, output buffers are required to strengthen the signals.

A new complementary CMOS logic style circuit, as shown in Fig. 3.8, is suggested for the implementation of the carry generator module for the full adder. It resembles the circuit of Fig. 3.7(c), except that an additional input c is introduced and the original inputs of a and b are rearranged. Comparing with the design of Fig. 3.7(c), this design lowers the switched capacitance of the preceding XOR/XNOR module due to the reduction of its fanouts from 2 to 1 for both the XOR and XNOR outputs. It is also easier to layout without the cross lines of $\overline{sel}$ and sel. Its robustness against voltage scaling and transistor sizing (high noise margins) enables it to operate reliably at low voltage and arbitrary (even minimal) transistor size.

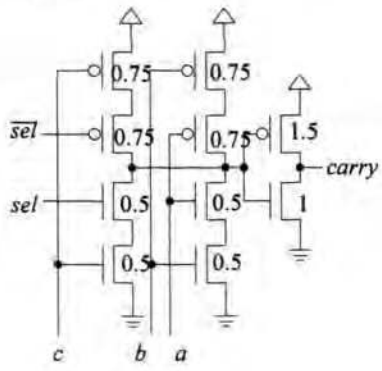


Figure 3.8 *Proposed carry generator module for full adder*

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

3.3.4 Circuit Structure of Hybrid 1 and Hybrid 2

The XOR/XNOR module of Fig. 3.4(d) in 6T, together with the XOR module of Fig. 3.6(b) and the carry generator module of Fig. 3.8 form a novel 1-bit full adder cell, abbreviated as Hybrid-1, which is shown in Fig. 3.9. The proposed XOR/XNOR module of Fig. 3.4(e), together with the XOR module of Fig. 3.6(b) and the carry generator module of Fig. 3.8 form another new 1-bit full adder cell, abbreviated as Hybrid-2, as shown in Fig. 3.10.

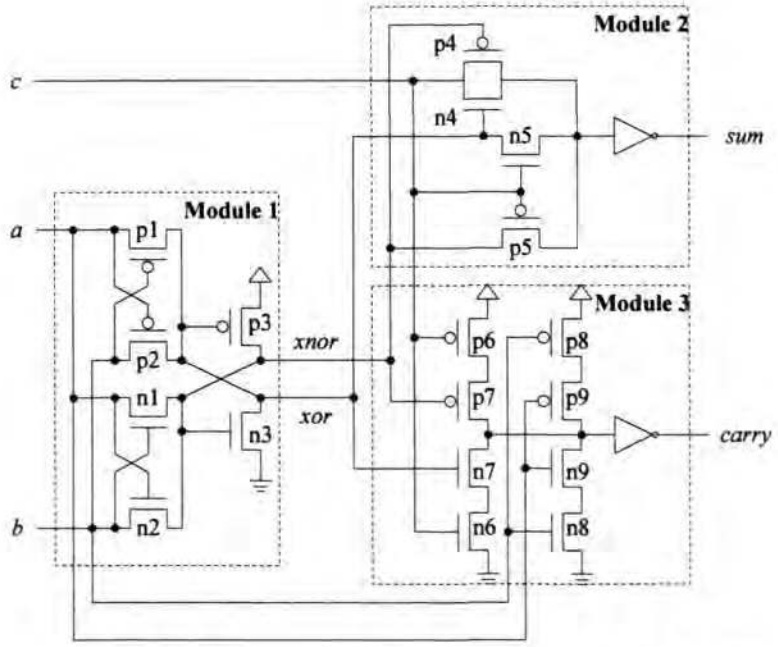


Figure 3.9 *Proposed 1-bit full adder circuit - Hybrid 1*

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

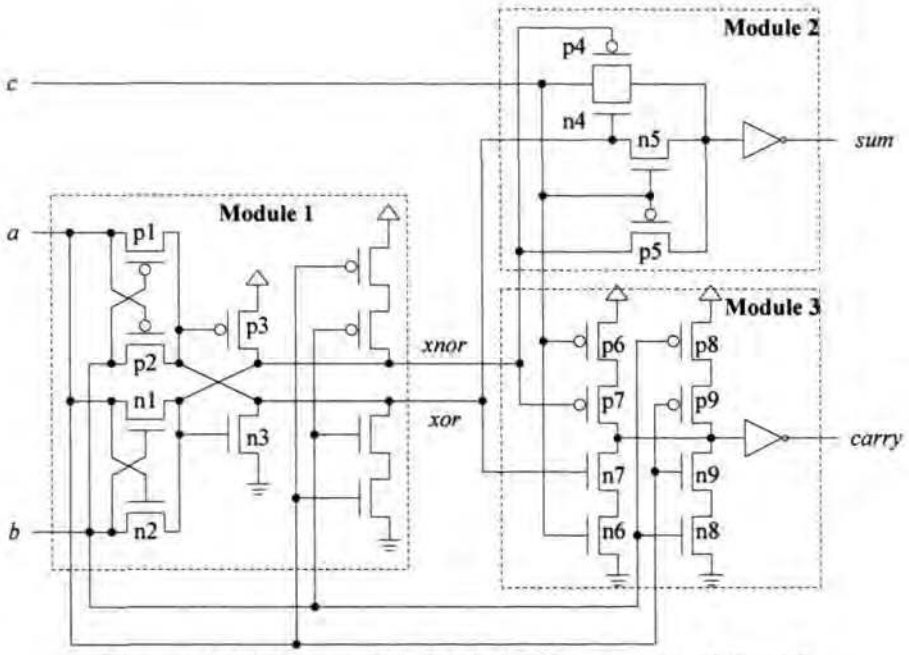


Figure 3.10 *Proposed 1-bit full adder circuit- Hybrid 2*

3.4 Simulation results

For a fair comparison, different full adders are optimized and simulated in a practical environment and with a unified optimization procedure based on the same process technology.

3.4.1 Simulation environment

It has been a common practice to treat the adder cell as a stand alone cell in the simulation [BUI02, RAD01, VES99(1), ZHU92]. It is also not unusual that the adder cells that perform well in such simulation still fail upon actual deployment because of the lack of driving power. This is because adder cells are normally cascaded to form a useful arithmetic circuit. Therefore, the adder cells must possess sufficient drivability to provide the next cell with clean inputs [SHA02]. In short, the driving cell must provide almost full-swing outputs to the driven cells. Otherwise the performance of the circuit will be degraded dramatically or become non-operative at low supply voltage. For this reason,

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

the adder cells of TFA, TGA, 14T and 10T cannot be cascaded without additional buffers attached to the outputs of each cell. This will be further verified later.

The authors of [SHA02] suggested one circuit structure, which is made of four cascaded adder cells, as shown in Fig. 3.11. This structure simulates the circuits like regular multipliers and binary adders that use full-adder cells as the building block. The inputs are fed from the buffers (two cascaded inverters) to give more realistic input signals and the outputs are loaded with buffers to give proper loading condition. All the required input-pattern-to-input-pattern transitions are included in the test patterns. The power consumption value is measured for the four cascaded adder cells, in addition to the intermediate buffers, while the delay is measured from the moment the inputs are applied to the first cell, until the latest of the *sum* and *carry* signals of the fourth cell is produced. However, this structure has some shortcomings. Firstly, although the first adder has exercised all the input-pattern-to-input-pattern transitions, the subsequent adders may not have all the input-pattern-to-input-pattern transitions exercised. Thus, it is not appropriate to consider the four cascaded cells as a whole and then divide the average power by four. As the last three adders are likely to consume lesser power than the first adder, this simulation tends to produce more optimistic power dissipation. Instead, it would be better to measure only the power dissipated by the first adder. Secondly, it is also noticed that every *carry* has two fan-outs while *sum* has only one fan-out. The loading of the two outputs is unbalanced.

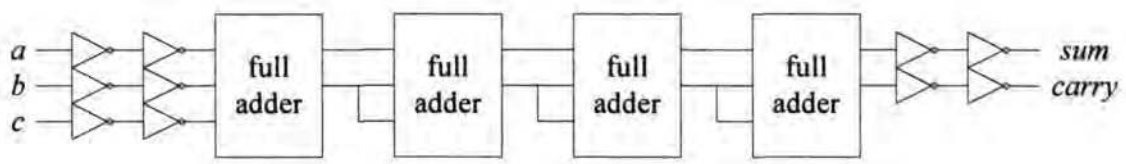


Figure 3.11 *Simulation setup suggested in* [SHA02]

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

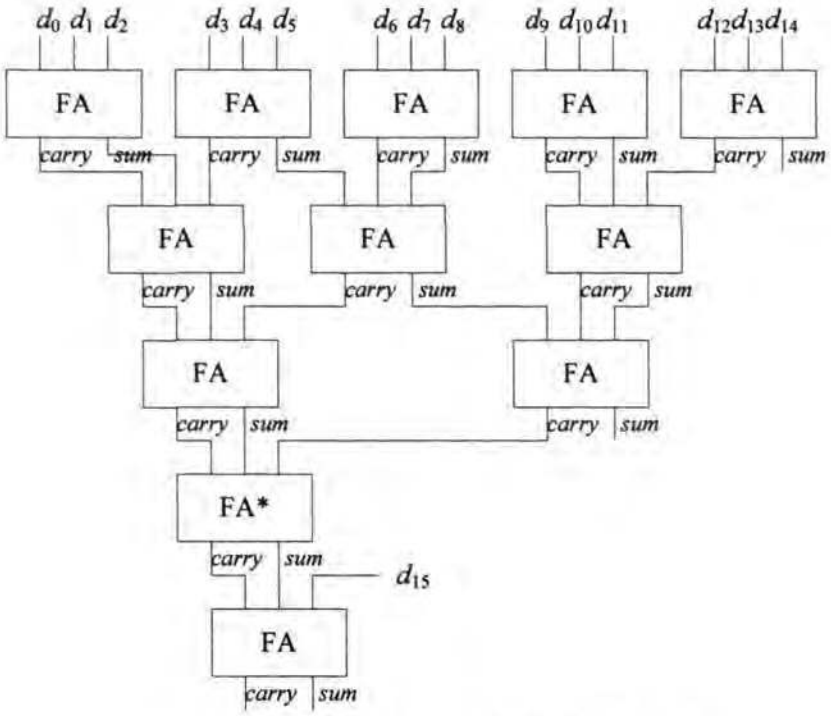


Figure 3.12 *Proposed simulation setup*

Our proposed simulation structure is shown in Fig 3.12, which emulates the tree structure of a parallel multiplier. Altogether 12 identical full adders are used, with the full adder (FA) marked with * being the cell of interest. The input signals of FA* are fed from the outputs of FA in the preceding stage, while the outputs of FA* are used to drive a FA in the following stage. This arrangement of full adders ensures that either the *carry* or *sum* output of each FA drives only one input of the FA in the next stage. The reason of cascading three levels of FAs preceding FA* is to examine the output drivability of the FA cells. If the FAs cannot provide enough driving power, the output signals after three successive stages will become very weak. Under this situation, FA* may fail to function.

The six circuits C-CMOS, CPL, TFA, TGA, 14T, 10T of Fig. 3.2 and our proposed adder cell, Hybrid-1 and Hybrid-2 are prototyped and simulated using the TSMC 0.18μm CMOS process with Level 49 technology file. The threshold voltages of the PMOS and NMOS transistors are around 0.46V and 0.48V, respectively. Star HSPICE is the circuit simulator used. For each simulation, HSPICE will generate an average power consumption value. The circuits are simulated at supply voltages range from 0.8 to 2.4V.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

The operating frequency is set at 100MHz. Backward derivation is performed to find a group of input test patterns which offers all the 56 different transitions from one input combination to another to FA* [SHA99]. For each transition, the delay is measured from 50% of the input voltage swing to 50% of the output voltage swing. The maximum delay of these 56 transitions is taken as the cell delay.

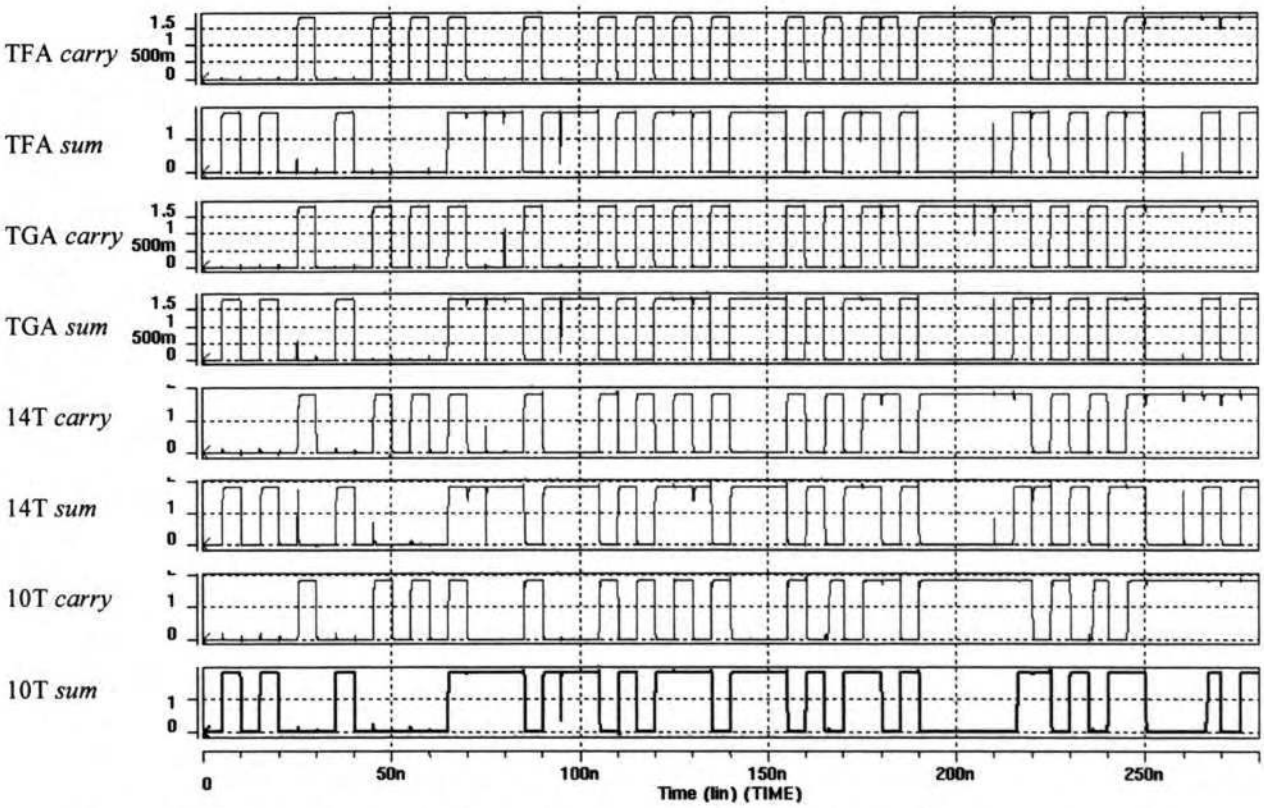


Figure 3.13 *Waveform snapshots of the circuits with attached buffers (1.8V, 100MHz)*

Buffers or inverters are attached to the TFA, TGA, 14T and 10T circuits to enhance their driving capability. Fig. 3.13 and Fig. 3.14 show the output waveforms of those four adders with and without the buffers added, respectively. The distorted output waveforms of Fig. 3.14 prove how important the drivability of adder cell is to the correct functionality of the circuit.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

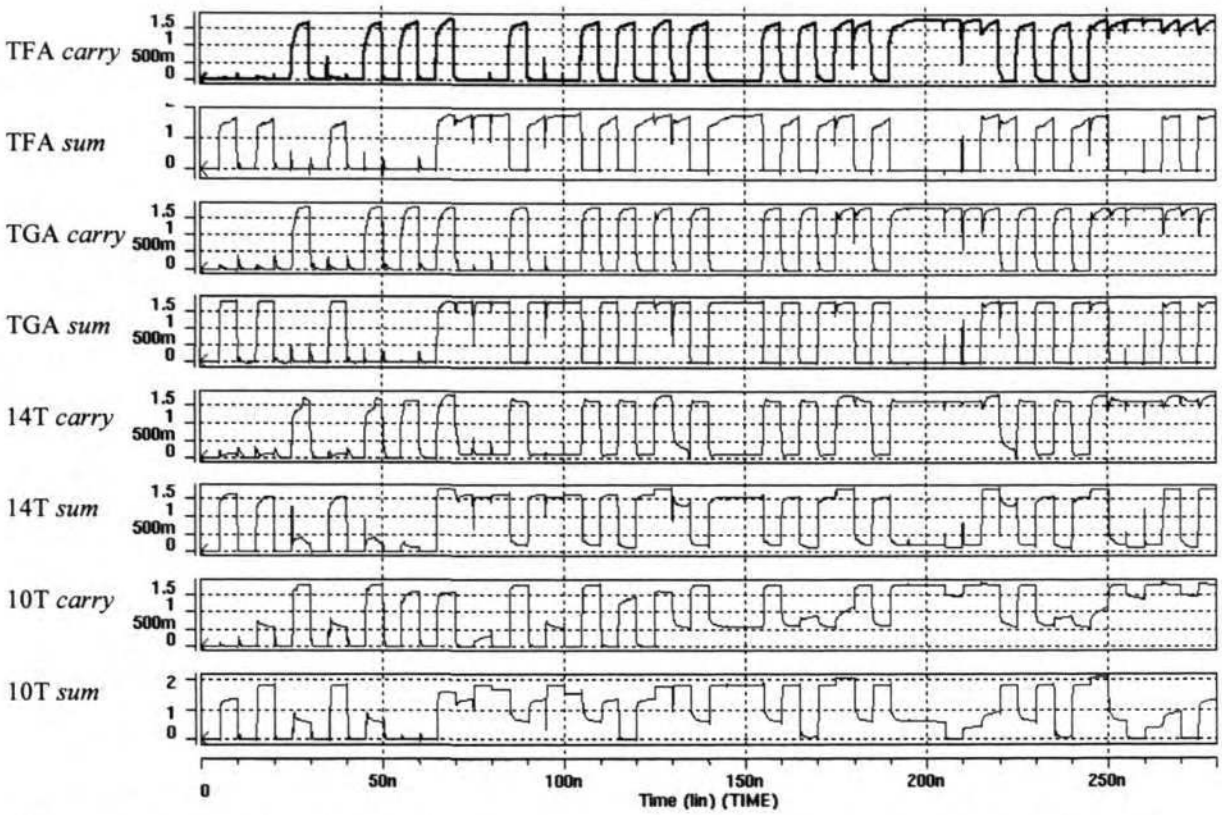


Figure 3.14 *Waveform snapshots of the circuits without attached buffers (1.8V, 100MHz)*

3.4.2 Transistor sizing optimization

As shown in [SAY02], the transistor sizing for optimal performance is technology dependent. The results of the six contender full adders are simulated under different process technologies in the literatures. In order to perform a fair comparison, we proposed an optimization procedure to size the transistor of all the full adders including these six reported full adders based on the same TSMC 0.18μm CMOS process technology.

The scaling operations are carried out in iterations transistor by transistor. To provide a good tradeoff between the somewhat conflicting power and delay performances, the goal of the optimization is to minimize the power-delay product (PDP). The power efficiency, or the power-delay product, measured in fJ is defined as the product of the worst-case delay and the average power consumption. This metric provides an indication of the

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

energy expended and the life span of the battery when the circuit is operating at its maximum speed. Suppose a circuit for optimization is composed of N transistors, labeled from T_1 to T_N and they are initialized with reasonable sizes at the outset. For a certain technology the channel lengths of all transistors are fixed at the minimal feature size, say 0.18μm in our example, so the only variable to be optimized is the channel width of each transistor. The first optimization run is begun with varying the channel width of T_1 in $2m+1$ steps and a step size of ψ to probe the circuit performance In other words, the different channel widths of T_1 simulated are $l_{1,0} - m\psi$, $l_{1,0} - (m-1)\psi$, …, $l_{1,0}$, …, $l_{1,0} + (m-1)\psi$, $l_{1,0} + m\psi$, where $l_{1,0}$ is the initial size of T_1. The probing sizes for T_1 are formally expressed as

$$l_{1,i} = l_{1,0} + i\psi \qquad \text{for } i = -m, -m+1, \ldots, 0, \ldots, m-1, m. \tag{3.4}$$

During this run, the sizes of all other transistors remaining unchanged.

Suppose that the jth channel width $l_{1,j}$ of T_1 provides the circuit with the lowest power-delay product through the simulation. We update T_1 with $l_{1,j}$ and carry on with the second run for T_2. After the second run, T_2 will be updated with its best channel width. The process goes on until the last transistor T_N is updated. An iteration is said to be completed when all the transistors have been updated. However, one iteration is not sufficient for the optimization because when a new transistor is sized in the current run, the other transistor sizes updated in the previous run may no longer maintain its optimality. Therefore, more iterations beginning with T_1 are needed. The iteration process stops when the performance difference in two successive iterations is smaller than a given error ε. Let Θ_{k-1} and Θ_k be the optimized power-delay product at the end of the $(k-1)$-th and k-th iterations, respectively. The termination criterion is given by:

$$\frac{\Theta_{i-1} - \Theta_i}{\Theta_i} \le \varepsilon \tag{3.5}$$

The flow chart of our proposed transistor sizing optimization procedure is shown in Fig. 3.15. In order to obtain enough coverage so that the optimal or quasi-optimal operating point would fall into the search region, and to allow for fine calibration, the resolution of

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

the sizing step ψ may be made variable. Large step size is used at the first few iterations and smaller step size is used for the remaining iterations.

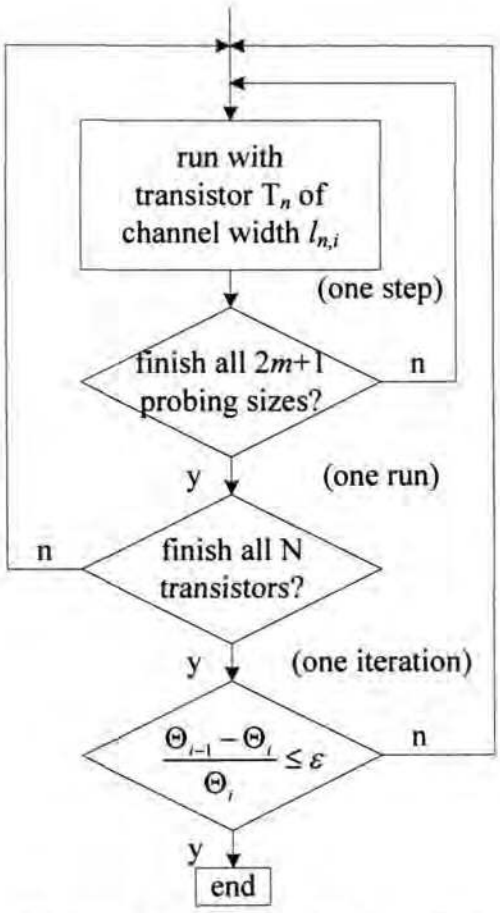


Figure 3.15 *Optimization flowchart*

In what follows, we will verify that our proposed optimization procedure is a convergent algorithm. In the above procedure, the sizing operation is carried out one transistor at a time such that the width of the transistor being tuned is set to a new value only when it improves the circuit's power-delay product. Therefore, the power-delay product at the end of one iteration will never be worse than that of the previous one, that is,

$$\Theta_k < \Theta_{k-1}. \tag{3.6}$$

Besides, all power-delay product values, including the power-delay product of the starting circuit, Θ_0 for a working circuit are finite non-negative numbers, that is

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

$$\Theta_k \geq 0. \tag{3.7}$$

From (3.6) and (3.7), we have

$$0 \leq \Theta_k \leq \Theta_0. \quad \text{(for any } k\text{)} \tag{3.8}$$

We define $\Delta_k = \Theta_{k-1} - \Theta_k$. From (3.6),

$$\Delta_k \geq 0. \tag{3.9}$$

Thus, we have

$$\begin{aligned} \Theta_K &= \Theta_0 - \sum_{k=1}^{K} (\Theta_{k-1} - \Theta_k) \\ &= \Theta_0 - \sum_{k=1}^{K} \Delta_k \end{aligned} \tag{3.10}$$

Its limit is given by

$$\lim_{K\to\infty} \Theta_K = \Theta_0 - \lim_{K\to\infty} \sum_{k=1}^{K} \Delta_k$$

If the proposed optimization procedure is not convergent, i.e., $\lim_{K\to\infty} \Theta_K$ diverges, which implies the divergence of $\lim_{K\to\infty} \sum_{k=1}^{K} \Delta_k$. From (3.9), the only possible situation for the divergence is

$$\lim_{K\to\infty} \sum_{k=1}^{K} \Delta_k \to \infty \tag{3.11}$$

which implies $\lim_{K\to\infty} \Theta_K \to -\infty$. This has violated (3.8). That establishes the claim that our optimization procedure is convergent.

In our optimization procedure, the starting sizes for the six previously reported full adders are the transistors' aspect ratio reported in [BUI02, CHA95, VES99(1), WES93, ZHU92]. For our proposed Hybrid-1 and Hybrid-2 cells, the initial sizes are estimated from standard practices and past experience. The step size of the first iteration in our example is set to 0.05μm, which is around 10% to 20% of the initial channel width. The step size of the subsequent iterations is reduced to 0.02μm. Thus, the final transistor sizes have the precision of 10% of the channel length, which is 0.18μm for our targeted technology. The iteration process termination error is set to 1%.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Two optimization strategies are adopted in the above procedure of transistor sizing to accelerate the process. Firstly, the corresponding PMOS and NMOS in a complementary pair are optimized in successive runs because the output transitions of the node driven by one transistor is often influenced most by the driving capability of its complementary counterpart. For example, the sizing of p2 and n2 in TGA are carried out in succession. Secondly, series transistors or parallel transistors of same type sourcing current to or sinking current from the same node are treated equally and can be optimized simultaneously. In C-CMOS, transistors p6, p7 and p8 are the parallel example while transistors p1 and p2 are the series example.

Optimization of the transistor sizing is carried out at two different voltages, 0.8V and 1.8V for C-CMOS, CPL, TFA, TGA and Hybrid-2. As Hybrid-1 and 14T can only work above 1.0V, these two cells are optimized at 1.0V and 1.8V. The lowest voltage that 10T can function under the proposed simulation setup is 1.8V, so it is optimized only at 1.8V. The final optimized transistor widths for each full adder cell are listed in Table 3.1.

Table 3.1 *Transistor sizes* (μm) *of full adders optimized for power-delay product*

(a) *C-CMOS*

	p1	**p2**	**p3**	**p4**	**p5**	**p6**	**p7**	**p8**	**p9**	**p10**	**p11**	**p12**
0.8V	0.78	0.78	0.8	0.8	0.8	1.2	1.2	1.2	0.54	0.68	0.68	0.68
1.8V	1.1	1.1	1.04	1.04	0.8	0.46	0.46	0.46	0.7	0.74	0.74	0.74
	n1	**n2**	**n3**	**n4**	**n5**	**n6**	**n7**	**n8**	**n9**	**n10**	**n11**	**n12**
0.8V	0.36	0.36	0.4	0.4	0.34	0.46	0.46	0.46	0.34	0.3	0.3	0.3
1.8V	0.4	0.4	0.46	0.46	0.3	0.4	0.4	0.4	0.5	0.3	0.3	0.3

(b) *CPL*

	p1	**p2**	**p3**	**p4**	**n1**	**n2**	**n3**	**n4**	**n5**	**n6**	**n7**	**n8**
0.8V	0.5	0.4	0.5	0.5	0.6	0.6	0.41	0.3	0.3	0.3	0.4	0.41
1.8V	0.4	0.4	0.4	0.4	0.55	0.55	0.45	0.55	0.5	0.5	0.7	0.35
	n9	**n10**	**n11**	**n12**	**n13**	**n14**	**n15**	**n16**	**n17**	**n18**	**n19**	**n20**
0.8V	0.37	0.37	0.3	0.44	0.44	0.58	0.38	0.38	0.4	0.4	0.4	0.3
1.8V	0.45	0.45	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

(c) *TFA*

	p1	**p2**	**p3**	**p4**	**p5**	**p6**	**p7**	**p8**
0.8V	0.57	0.6	0.8	0.3	0.45	0.4	0.3	0.3
1.8V	0.31	0.78	0.31	0.33	0.45	0.3	0.7	0.32
	n1	**n2**	**n3**	**n4**	**n5**	**n6**	**n7**	**n8**
0.8V	0.35	0.35E	0.35E	0.35	0.5	0.32	0.3	0.49
1.8V	0.3	0.3	0.68	0.3	0.57	0.3	0.3	0.45

(d) *TGA*

	p1	**p2**	**p3**	**p4**	**p5**	**p6**	**p7**	**p8**	**p9**	**p10**
0.8V	0.3	0.32	0.88	0.51	1.01	0.32	0.6	0.3	0.61	0.34
1.8V	0.34	0.35	0.78	0.4	0.65	0.3	0.44	0.3	0.5	0.4
	n1	**n2**	**n3**	**n4**	**n5**	**n6**	**n7**	**n8**	**n9**	**n10**
0.8V	0.3	0.3	0.31	0.3	0.3	0.3	0.3	0.3	0.3	0.3
1.8V	0.31	0.34	0.3	0.3	0.3	0.3	0.32	0.55	0.3	0.3

(e) *14T*

	p1	**p2**	**p3**	**p4**	**p5**	**p6**	**p7**
1.0V	1.85	1.85	0.88	1.75	0.75	1.7	1.8
1.8V	1.7	1.7	0.6	1.7	0.38	1.5	1.7
	n1	**n2**	**n3**	**n4**	**n5**	**n6**	**n7**
1.0V	0.3	0.3	0.3	0.36	1.62	0.75	0.4
1.8V	0.7	0.7	0.3	0.4	1.5	0.4	0.4

(f) *10T*

	p1	**p2**	**p3**	**p4**	**p5**
1.8V	0.95	0.78	0.7	0.7	0.6
	n1	**n2**	**n3**	**n4**	**n5**
1.8V	1.03	0.7	0.7	0.7	0.7

(g) *Hybrid-1*

	p1	**p2**	**p3**	**p4**	**p5**	**p6**	**p7**	**p8**	**p9**
1.0V	0.9	0.9	0.3	0.4	0.3	0.7	0.7	0.5	0.5
1.8V	0.71	0.71	0.65	0.3	0.35	0.5	0.5	0.54	0.54
	n1	**n2**	**n3**	**n4**	**n5**	**n6**	**n7**	**n8**	**n9**
1.0V	0.32	0.32	0.3	0.4	0.55	0.38	0.38	0.32	0.32
1.8V	0.3	0.3	0.3	0.3	0.3	0.4	0.4	0.4	0.4

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

(h) *Hybrid-2*

	p1	**p2**	**p3**	**p4**	**p5**	**p6**	**p7**	**p8**	**p9**	**p10**	**p11**
0.8V	0.33	0.33	0.4	0.6	0.6	0.31	0.3	0.58	0.58	0.5	0.5
1.8V	0.42	0.42	0.31	0.8	0.8	0.4	0.3	0.7	0.7	0.5	0.5
	n1	**n2**	**n3**	**n4**	**n5**	**n6**	**n7**	**n8**	**n9**	**n10**	**n11**
0.8V	0.3	0.3	0.26	0.3	0.3	0.31	0.56	0.3	0.3	0.3	0.3
1.8V	0.6	0.6	0.25	0.3	0.3	0.3	0.3	0.4	0.4	0.3	0.3

3.4.3 Simulation results and analysis

The pre-layout simulation of power, delay and power-delay product plotted against supply voltage ranges from 0.8V to 2.4V of the Hybrid-1, Hybrid-2, C-CMOS, CPL, TFA, TGA, 14T, 10T are shown in Table 3.2 for comparison. The transistor sizes of Table 3.1 optimized at 0.8V and 1V are used for the simulation at the lower supply voltage range of 0.8V to 1.2V and the transistor sizes optimized at 1.8V are used for the simulation at the higher supply voltage range.

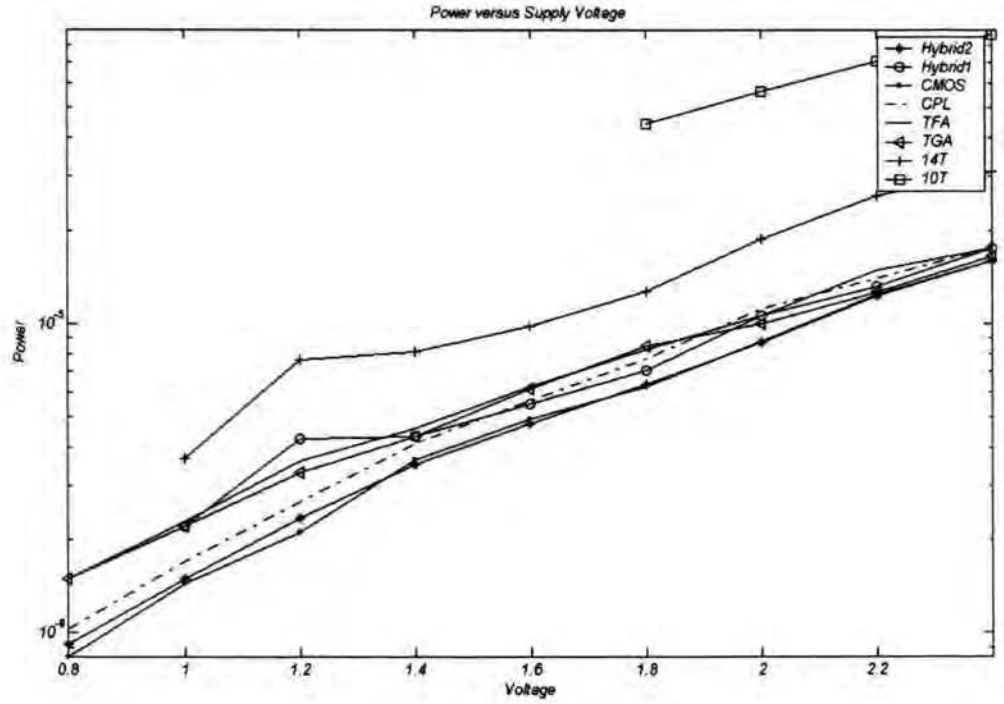
ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Table 3.2 *Power, delay and power-delay-product comparison of full adder cells*

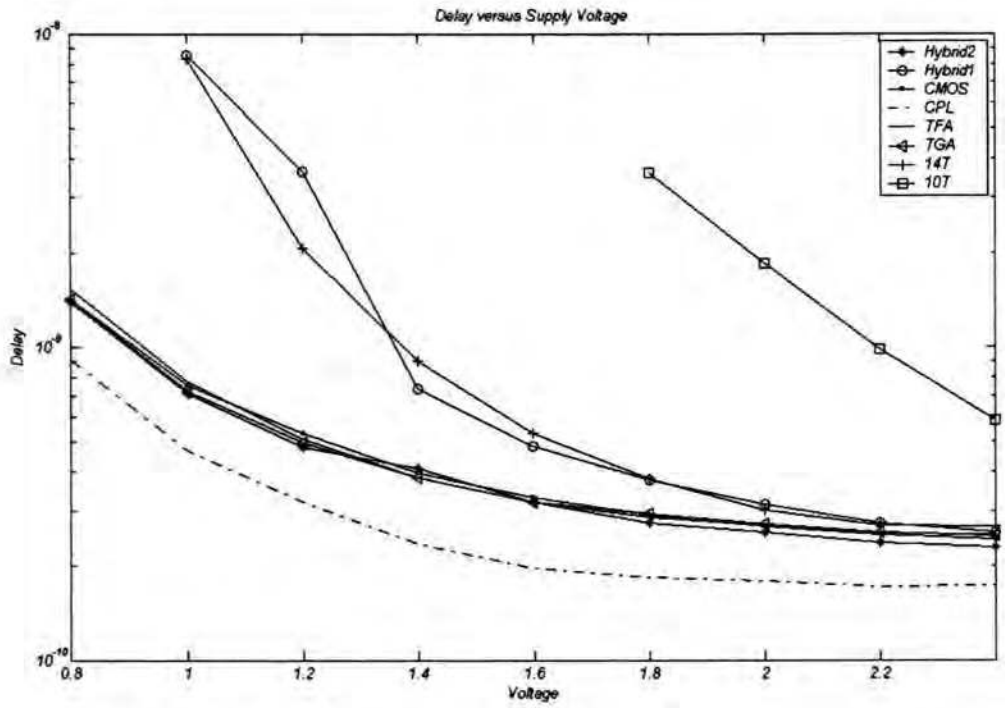
Vcc(V)	0.8	1.0	1.2	1.4	1.6	1.8	2.0	2.2	2.4
Power (μW)									
Hybrid-1		2.20	4.22	4.32	5.47	7.09	10.6	13.2	17.6
Hybrid-2	0.918	1.50	2.35	3.49	4.75	6.39	8.71	12.3	15.9
C-CMOS	0.84	1.45	2.12	3.63	4.91	6.23	8.77	12.4	15.9
CPL	1.03	1.70	2.64	4.08	5.64	7.72	11.2	14.0	17.7
TFA	1.50	2.28	3.60	4.56	6.25	8.25	10.6	14.9	17.6
TGA	1.49	2.20	3.30	4.29	6.12	8.47	10.0	12.6	16.5
14T		3.66	7.62	8.14	9.82	12.7	18.8	26.0	31.0
10T						44.4	56.5	70.7	86.4
Delay (ns)									
Hybrid-1		8.56	3.63	0.734	0.484	0.376	0.316	0.276	0.256
Hybrid-2	1.40	0.708	0.479	0.412	0.320	0.275	0.256	0.239	0.231
C-CMOS	1.42	0.756	0.531	0.397	0.333	0.292	0.269	0.252	0.244
CPL	0.908	0.468	0.321	0.236	0.197	0.184	0.179	0.172	0.173
TFA	1.53	0.777	0.511	0.385	0.322	0.288	0.270	0.255	0.252
TGA	1.42	0.721	0.497	0.383	0.321	0.294	0.274	0.257	0.250
14T		8.30	2.06	0.902	0.531	0.382	0.303	0.271	0.268
10T						3.61	1.85	0.986	0.584
Power Delay Product (fJ)									
Hybrid-1		18.83	15.32	3.171	2.647	2.666	3.350	3.643	4.506
Hybrid-2	1.285	1.062	1.126	1.438	1.520	1.757	2.230	2.940	3.673
C-CMOS	1.193	1.096	1.126	1.441	1.635	1.819	2.359	3.125	3.880
CPL	0.935	0.796	0.847	0.963	1.111	1.420	2.005	2.408	3.062
TFA	2.295	1.772	1.840	1.756	2.012	2.376	2.862	3.799	4.435
TGA	2.116	1.586	1.640	1.643	1.965	2.490	2.740	3.238	4.125
14T		31.33	27.66	5.97	4.75	4.78	5.94	7.18	7.94
10T						160.3	104.5	69.7	50.5

Fig.3.16 shows the visualized comparison based on the data in Table 3.2.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

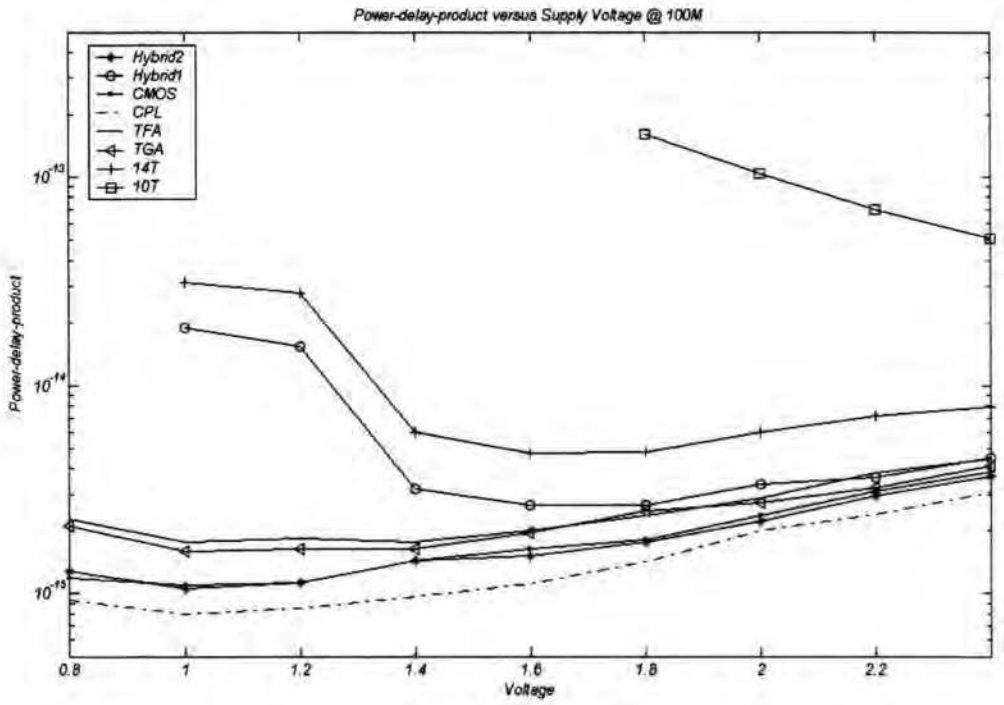


(a) *Average power performance*



(b) *Worst-case delay performance*

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library



(c) *Power-delay product*

Figure 3.16 *Power, delay and power-delay-product comparison of full adder cells*

The simulation results show that the 10T adder cell fails to function at low voltage. The lowest voltage it can operate at 100MHz is 1.8V. The excessive power dissipation and long delay are attributed to the threshold voltage drop problem and the poor driving capability of some internal nodes at input combinations that create non full-swing transitions.

The speed of the 14T decreases faster with the supply voltage than other adder cells, so does its power-delay product. The XOR/XNOR module circuit of 14T is the same as that of Hybrid-1 cell, so both of them suffer from the same threshold voltage drop problem that has been discussed in Section 3.3. Although Hybrid-1 cell has higher transistor count, it outperforms 14T in terms of power and power-delay-product, due to the merits of the XOR and carry generator modules. In addition, 14T fails to function at 0.8V.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

The simulation results also show that C-CMOS, TFA, TGA and CPL and Hybrid-2 still can work reliably when the supply voltage has dropped to 0.8V. Although TFA and TGA have lesser transistor count, due to the lack of drivability, additional buffers are required at each output, which increased their short-circuit power as well as switching power.

It is shown that Hybrid-2 and C-CMOS full adder are the most power efficient cells. Hybrid-2 is slightly faster than C-CMOS and as a result, it exhibits smaller power-delay-product than C-CMOS except at very low voltage of 0.8V. Due to the decoupling of the input and output circuitries, good drivability can be achieved by the proposed Hybrid cell. The full swing restoring transistors in Module 1 of the Hybrid cell overcome the weak logic problem, allowing it to operate reliably down to an ultra low voltage of 0.8V.

Although Hybrid-2 has four more transistors than Hybrid-1, it does not consume more power than Hybrid-1. This is because the elimination of the threshold voltage drop problem has allowed full voltage swing at the intermediate nodes, XOR and XNOR. Thus, spurious transitions and unnecessary switching activities are reduced in the XOR and carry generator modules. Therefore, the overall power consumption of the full adder cell is lower. Since Hybrid-2 cell has better power and delay performance than Hybrid-1 cell, it is not surprising that it is more energy-efficient than Hybrid-1. This is particularly prominent at low supply voltage.

C-CMOS works reliably at all voltage range especially at very low voltage. However, as already mentioned in Section 3.2, it generates *sum* using *carry* signal as the input, which causes the undesirable additional delay. This delay skew also leads to more spurious transitions to the cascaded stage. We define a normalized delay difference factor δ:

$$\delta = \frac{t_{max} - t_{min}}{t_{max}} \tag{3.12}$$

where t_{max} is the larger delay value of the *sum* and *carry* outputs, and t_{min} is the smaller delay value of the two output signals.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

The δ values of Hybrid-2 cell and C-CMOS cell are plotted against the supply voltage in Fig. 3.17. It is shown that the outputs of Hybrid-2 cell are more balanced than those of C-CMOS especially at low voltage.

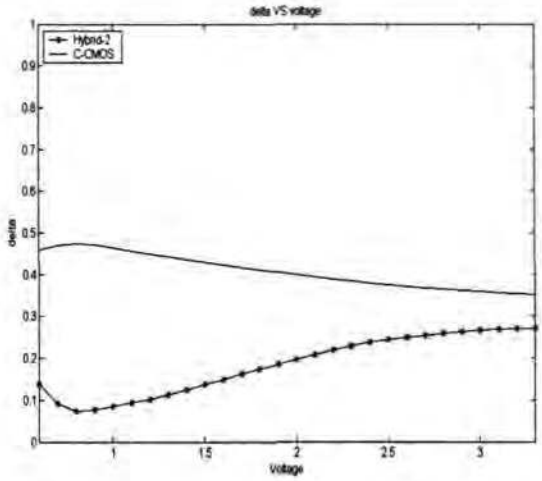


Figure 3.17 *Normalized delay difference factors between Hybrid-2 and C-CMOS*

Fig. 3.18 shows a snapshot of the sum and carry outputs of Hybrid-2 and C-CMOS full adder cells. Due to the unbalanced output of C-CMOS adder, its outputs generate more glitches than those of Hybrid-2 cell for the same input transitions.

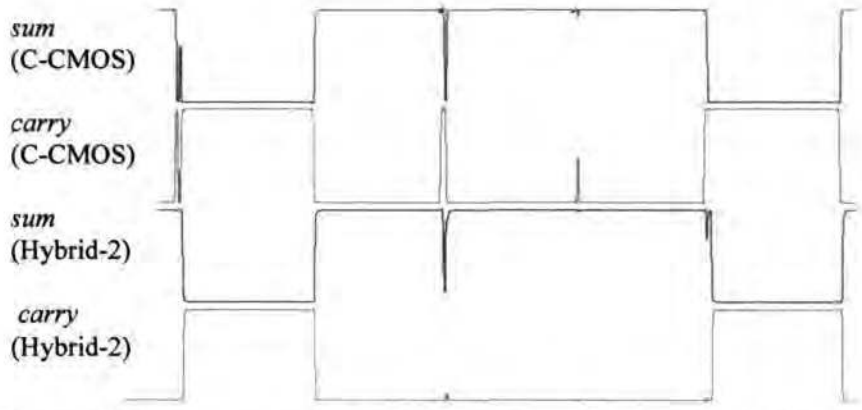


Figure 3.18 *Comparison of output waveforms between Hybrid-2 and C-CMOS*

The skew of the sum and carry outputs causes the carry-save outputs of the Wallace/Dadda tree [WAN95] to have irregular input arrival profile. Although the fast

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

carry can be consumed in the critical path of the adder in the next stage and the slower sum be consumed in the faster path of the next stage for constructing an optimized adder tree [OKL96], such construction will annihilate the flexibility to re-distribute the adder cells to maximize the area efficiency and to eliminate cross-stage interconnections described by the technique presented in [WAN95]. Therefore, if the length of wiring and area efficiency are prominent cost function elements, which is the case for large arithmetic circuit in ultra deep submicron technology, balanced delays without jeopardizing the worst case delay can be a desirable attribute.

Despite being the fastest circuit, CPL consumes higher power than Hybrid-2 and C-CMOS because of its dual-rail structure and the substantial number of internal nodes. The additional inverters used to generate the complement inputs have also increased the power consumption. This excessive overhead offsets the advantages of efficient XOR realization offered by this logic design style. Although CPL has achieved an overall good power-delay product due to its excellent speed, the performance is highly sensitive to the transistor scaling as observed in the optimization process described in Section 3.4.2. Besides, the need to generate complementary signals for all the surrounding circuits also limits the usage of CPL.

The complete layout of the Hybrid-1 cell is shown in Fig. 3.19. On top of cleverly exploring the complementary outputs of the XOR/XNOR module to produce a full-swing *carry*, the new circuit of carry generator has a very regular structure. Comparing with the layouts of the other two modules, both of which have only 6 transistors, the area saved by the reduced layout complexity of the carry generator module has compensated for its increase in transistor count (10 transistors).

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

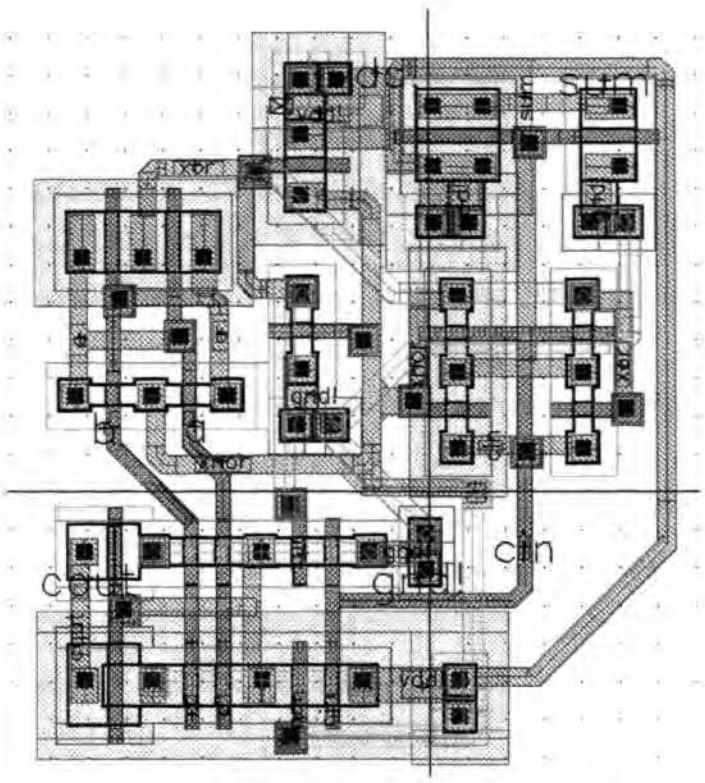


Figure 3.19 *Layout of the Hybrid-1 cell*

The layouts of the Hybrid-2 cell are shown in Fig 3.20. Comparing with the Hybrid-1 cell, the layout of its Module 1 is more regular and fits nicely with the other two modules in a complete full adder cell.

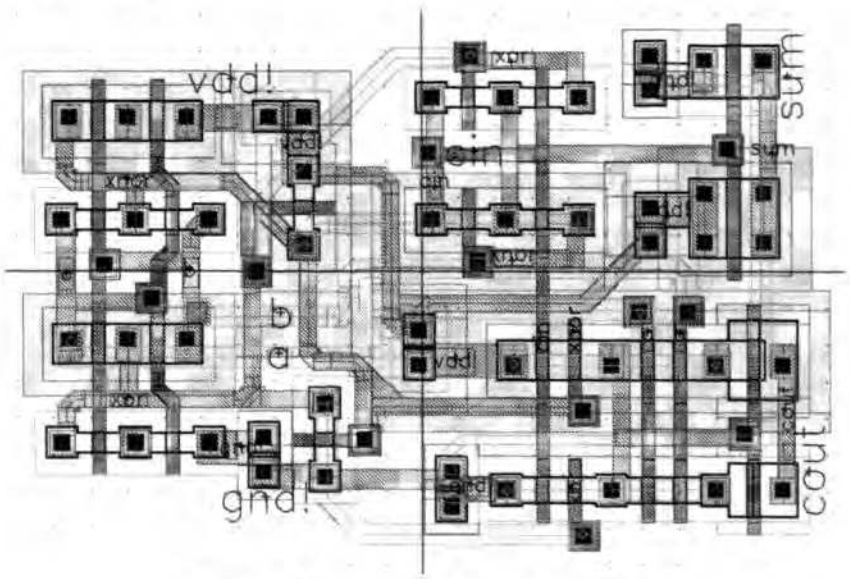


Figure 3.20 *Layout of the Hybrid-2 cell*

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

All full adder cells are laid out with optimized sizing and spacing in compliance to the design rules of TSMC 0.18 μm CMOS process. The values of the length, width and overall area of the adder cells are listed in Table 3.3. The layouts of CPL and TGA full adders occupy the most silicon area. CPL adder needs more metal lines to connect the complementary inputs. Besides, the style of the transistor connection of CPL is not suitable for area-efficient layout. TGA adder is composed of transmission gates, which occupies more area due to the inefficient usage of the n-type wells. The layout of the Hybrid-2 cell occupies a much smaller silicon area, which is less than 60% of the area of the CPL. It is only slightly larger than 14T and 10T adders. The area ascendancy of 14T and 10T is due primary to their smaller number of transistors, but the area gained from the reduced transistor count is offset by the penalty of the irregular circuit structure of pass transistors. Besides, their overall performances are also inferior.

Table 3.3 *The silicon area of the full adders*

	Hybrid-1	Hybrid-2	C-CMOS	CPL	TFA	TGA	14T	10T
Length (μm)	9.67	7.18	17.37	11.20	9.58	14.07	12.10	11.2
Width (μm)	8.415	10.97	5.76	12.20	10.08	9.59	5.64	6.30
Area (μm^2)	81.373	78.76	100.05	136.64	96.57	134.88	68.24	70.56

The post layout simulation includes the parasitic information extracted from both the transistors as well as the interconnections. Due to the scale of the 1-bit full adder circuits, the power, delay and power-delay product of the extracted circuits (RC extraction) do not deviate much from those of the pre-layout circuits, especially for the Hybrid-2 cell. This is because of its regular structure and shorter wire length. The comparisons of the pre- and post-layout performance at some standard supply voltages are shown in Table 3.4.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Table 3.4 *The pre- and post-layout simulation results of the Hybrid-1 and Hybrid-2 cells*

		Hybrid-1 (pre)	Hybrid-1 (post)	Hybrid-2 (pre)	Hybrid-2 (post)
2.4V	Power (μW)	17.63	19.24	15.92	16.32
	Delay (ps)	256.02	252.41	231.43	234.98
	PDP (fJ)	4.51	4.86	3.68	3.83
1.8V	Power (μW)	7.090	7.811	6.392	6.694
	Delay (ps)	37.64	36.28	27.53	27.84
	PDP (fJ)	2.67	2.83	1.76	1.86
1.0V	Power (μW)	2.20	2.46	1.50	1.47
	Delay (ps)	856.31	950.86	708.35	675.81
	PDP (fJ)	1.88	2.34	1.06	0.99

3.5 Summary

For full adder cell design, pass-logic circuit is thought to be dissipating minimal power and have smaller area because it uses less number of transistors. Thus, CPL adder is considered to be able to perform better than CMOS adder in [ZIM97]. However, in our opinion, pass-logic circuit usually has irregular structure, which increases the wiring complexity and its performance is highly susceptible to transistor sizing. On the other hand, the complementary CMOS logic circuit has the advantages of layout regularity and stability at low voltage. Therefore, it is the different design constraints imposed by the applications that each logic design style has its place in the cell library development.

An optimization procedure is proposed to size the transistor of the full adder cells in order to perform a fair comparison because the transistor sizing for optimal performance is technology dependent. The optimization procedure takes sweeping method to search the optimized transistor sizes in iterations, which has been proved a convergent algorithm.

In the past, full adders are often evaluated in isolation without concern on how they are deployed in actual circuit [BUI02, RAD01, VES99(1), ZHU92]. We argue that a 1-bit full adder cell that functions correctly in stand alone simulation is not sufficient to validate its actual performance or even functionality when it is integrated into a larger circuit. Inadequate consideration in simulation setup tends to produce the level of

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

performance optimistically above the capability of the circuit being simulated. In this chapter, we proposed a reasonably simple architecture to simulate the adder cell in an environment realistic to its actual deployment in the most frequently used parallel multiplier structure.

In summary, two novel 1-bit adder cells consisting of the XOR/XNOR, sum and carry out subcircuits, are proposed. The pass logic design style is used to efficiently generate the XOR and XNOR functions simultaneously and a good drivability carry output is generated by a novel complementary CMOS style circuit with regular layout. In addition, the last-stage inverter de-couples the output and input to improve the driving capability. Despite having higher transistor count than the recently reported designs, the two cells have shown to be highly power efficient over a wide supply voltage range. The balanced sum and carry outputs also offer considerable flexibility in allocating the adder cells in tree structured circuit to eliminate as many cross-stage interconnections and to reduce the maximum length of in-stage interconnections without affecting the critical path delay as there is no discrimination on any port for any legitimate connections [WAN95]. Although its power-delay performances is comparable to C-CMOS and poorer than CPL, the area efficient layout made it a good choice for implementing large tree structured arithmetic circuit when the overall performance and area efficiency are prominent cost function elements.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Chapter 4

Low Voltage, Low Power 4-2 and 5-2 Compressors for Fast Arithmetic Circuits

4.1 Introduction

In the last chapter, we explore different CMOS logic styles to the design and performance evaluation of low voltage, low power full adder cells. These primitive full adder cells are also referred to as 3-2 counters or compressors when they are used to compress the columns of the partial product matrix in the CSA tree of parallel multiplier. To reduce the number of stages in the CSA tree, 4-2 and 5-2 compressors are frequently employed in large high-speed multiplier. This chapter further presents several architectures and designs of low power 4-2 and 5-2 compressors capable of operating at ultra-low supply voltage. The architectures of these compressors are anatomized into their constituent modules and different static logic styles based on the same advanced submicron CMOS process model are used to realize them. Different configurations of each architecture, which include a number of novel 4-2 and 5-2 compressor designs, are prototyped and simulated to evaluate their performance in speed, power dissipation and power-delay product. The newly developed circuits are based on various configurations of the novel 5-2 compressor architecture with the new carry generation circuit, or existing architectures configured with the proposed circuit for the XOR-XNOR module. Driving capability has been considered in the design as well as in the simulation setup so that these 4-2 and 5-2 compressor cells can operate reliably in any tree structured parallel multiplier at very low supply voltage. Two new simulation environments are created to ensure that the performances reflect the realistic circuit operation in the system to which these cells are

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

integrated. Simulation results show that the 4-2 compressor with the proposed XOR-XNOR module and the new fast 5-2 compressor architecture are able to function at supply voltage as low as 0.6V, and outperform many other architectures including the classical complementary CMOS logic compressors and variants of compressors constructed with various combinations of recently reported superior low-power logic cells [MAR99, PRA01, RAD00, KWO00].

The rest of this chapter is organized as follows. Section 4.2 investigates the performances of several fast 4-2 compressors. The 4-2 compressors constructed around the proposed XOR-XNOR cell in Chapter 2 exhibit superior power efficiency compared with other configurations of the same architecture. Section 4.3 scrutinizes the 5-2 compressor architectures and their underlining building modules. A new fast 5-2 compressor architecture is proposed, together with a new circuit for its carry generation module. This new architecture performs well with almost any configuration of logic styles and its overall performance is the best among the known 5-2 compressor architectures. Special circuits for the simulation of the 4-2 and 5-2 compressors are described in Section 4.4. These setups emulate a realistic application environment that truly reflects its actual operability in a tree-structured multiplier. The simulation results are given and analyzed. Finally, the summary in Section 4.5 concludes the chapter.

4.2 4-2 compressor

A 4-2 compressor has five inputs and three outputs, as shown in Fig. 4.1. The four inputs x_1, x_2, x_3 and x_4, and the output, *sum* have the same weight. The output, *carry* is weighted one binary bit order higher. The 4-2 compressor receives an input, c_{in} from the preceding module of one binary bit order lower in significance, and produces an output c_{out} to the next compressor module of higher significance. Different structures of 4-2 compressors exist and they all have to abide by the fundamental equation (4.1):

$$x_1 + x_2 + x_3 + x_4 + c_{in} = sum + 2 \cdot (carry + c_{out}) \tag{4.1}$$

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Besides, to accelerate the carry save summation of the partial products, it is imperative that the output, c_{out} be independent of the input, c_{in}.

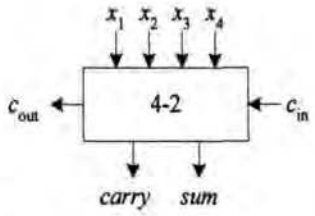


Figure 4.1 *4-2 compressor*

The conventional implementation of a 4-2 compressor is composed of two serially connected full adders, as shown in Fig. 4.2. At gate level, high input compressors are anatomized into XOR gates and carry generators normally implemented by multiplexers (MUX). Therefore, different designs can be classified based on the critical path delay in terms of the number of primitive gates. Let Δ_{XOR} denote the delay of an XOR gate and Δ_{CGEN} denote the delay of a carry generator. A compressor is said to have a delay of $(m\Delta_{XOR} + n\Delta_{CGEN})$ if its critical path consists of m XOR gates and n carry generators. Since the difference between the delays of widely used XOR gate and carry generator is trivial in an optimized design, the delay of the compressor is more commonly specified as $(m + n)\Delta$. Therefore, the straightforward implementation of a 4-2 compressor of Fig. 4.2 has a long critical path delay of 4Δ [HSI98]. Additionally, due to the uneven delay profiles of the outputs arriving from different input paths, the carry save adder (CSA) tree for the partial product accumulation constructed from such cells tends to generate a lot of glitches.

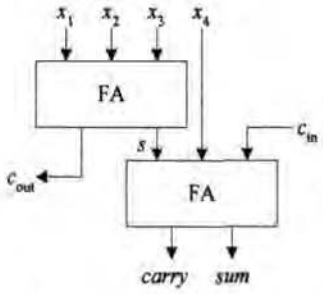


Figure 4.2 *Conventional 4-2 compressor - 4Δ*

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

A 4-2 compressor flattened and optimized at gate level to reduce the critical path delay is shown in Fig. 4.3 [RAD00]. It uses more than 80 transistors when implemented in conventional or complementary CMOS logic style. As this circuit is capable of functioning below 1V, it is used as a benchmark for evaluating the performance of other low voltage and low power 4-2 compressor circuits.

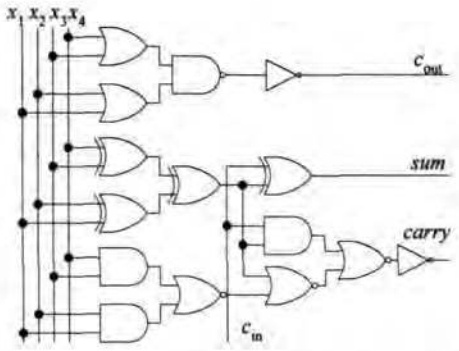


Figure 4.3 *Logic level optimized CMOS 4-2 compressor*

A recent design of 4-2 compressor [GU03(1), MAR99, PRA01, RAD00] is derived from the modified equations for the functions of Fig. 4.2. The three outputs of the design are described by (4.2), (4.4) and (4.5) as follows:

$$c_{out} = (x_1 \oplus x_2)\cdot x_3 + x_1 \cdot x_2 = (x_1 \oplus x_2)\cdot x_3 + \overline{(x_1 \oplus x_2)}\cdot x_1 \tag{4.2}$$

$$s = x_1 \oplus x_2 \oplus x_3 \tag{4.3}$$

$$sum = s \oplus x_4 \oplus c_{in} = x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus c_{in} \tag{4.4}$$

$$\begin{aligned} carry &= (s \oplus x_4)\cdot c_{in} + s \cdot x_4 \\ &= (x_1 \oplus x_2 \oplus x_3 \oplus x_4)\cdot c_{in} + \overline{(x_1 \oplus x_2 \oplus x_3 \oplus x_4)}\cdot x_4 \end{aligned} \tag{4.5}$$

The two carry signals, *carry* and c_{out}, are generated from both the Exclusive OR (XOR) and Exclusive NOR (XNOR) functions of the input signals. The *sum* output is generated by several two-input XOR circuits, some internal signals of which can be used to generate the two carries. Fig. 4.4 shows the logic decomposition of this 4-2 compressor architecture. It is mainly composed of six modules, four of which are XOR circuits and the other two are 2-1 multiplexers. Three special XOR-XNOR modules marked with

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

"XOR*" generate both the XOR and XNOR signals simultaneously to other modules driven by them. This design has a critical path delay of 3Δ, which is 1Δ delay shorter than the conventional implementation. Besides, its outputs feature balanced signal arrival time from each data inputs (x_1 to x_4), thanks to the special "XOR*" modules.

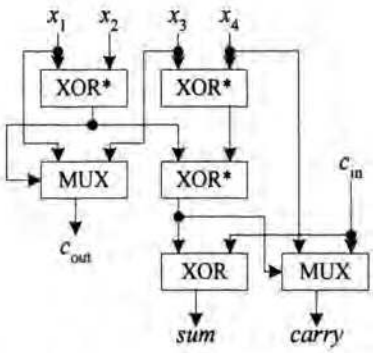


Figure 4.4 *Logic decomposition of 4-2 compressor - 3Δ*

As both the XOR and XNOR signals are required in the carry generation circuits of Fig. 4.4, circuits capable of co-generating these two signals, as depicted in Fig. 3.4 in Chapter 3, are beneficial to the implementation of the special XOR* modules. Although there is no need for the XOR* module with inputs of x_3 and x_4 to provide the XNOR output, the same structure as other XOR* modules is still preferred in order to avoid skewed delay paths.

The layout of the proposed new 4-2 compressor is shown in Fig. 4.5. It is based on the novel XOR/XNOR cell of Fig. 3.4(e), the XOR circuit of Fig. 3.6(b) for the generation of the sum, and the carry generation circuit of Fig. 3.7(c) from Chapter 3. It occupies a silicon area of 22μm×17μm.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

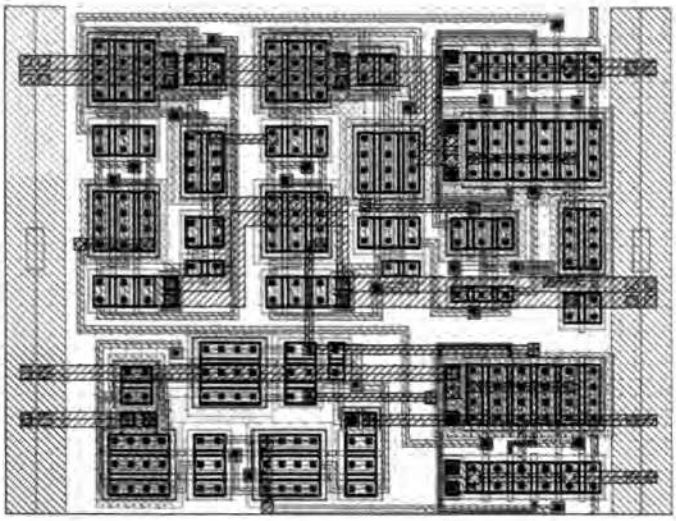


Figure 4.5 *Layout of the new 4-2 compressor using the proposed XOR-XNOR cell*

4.3 5-2 compressor

4.3.1 5-2 compressor architectures

5-2 compressor is another widely used building block for high precision and high speed multipliers. The block diagram of a 5-2 compressor is shown in Fig. 4.6, which has seven inputs and four outputs. Five of the inputs are the primary inputs x_1, x_2, x_3, x_4 and x_5, and the other two inputs, c_{in1} and c_{in2} receive their values from the neighboring compressor of one binary bit order lower in significance. All the seven inputs have the same weight. The 5-2 compressor generates an output, *sum* of the same weight as the inputs, and three outputs, *carry*, c_{out1} and c_{out2} weighted one binary bit order higher. The outputs, c_{out1} and c_{out2} are fed to the neighboring compressor of higher significance. All the 5-2 compressors of different designs abide by the generic equation (4.6):

$$x_1 + x_2 + x_3 + x_4 + x_5 + c_{in1} + c_{in2} = sum + 2 \cdot (carry + c_{out1} + c_{out2}) \quad (4.6)$$

Besides, to speedup the carry save summation of the partial products, the output c_{out1} must be independent of the inputs, c_{in1} and c_{in2}, and the output, c_{out2} must be independent of the input, c_{in2}.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

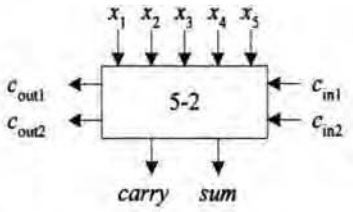


Figure 4.6 *5-2 compressor*

A simple implementation of the 5-2 compressor is to cascade three full adders in a hierarchical structure, as shown in Fig. 4.7, which has a critical path delay of 6Δ.

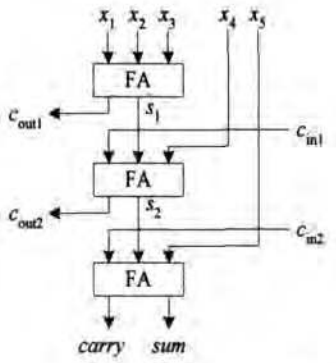


Figure 4.7 *5-2 compressor based on cascaded full adders - 6Δ*

Fig. 4.8 shows a modified architecture of 5-2 compressor [PRA01], which has a critical path delay of 5Δ. The XOR* modules in the figure generate both the XOR and XNOR signal simultaneously, as described in Chapter 3. The style and structure of the circuit share some common attributes as the recently published structural design of full adders [PRA01, RAD01, SHA97, SHA00, SHA02] and 4-2 compressors [GU03(1), PRA01, RAD00].

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

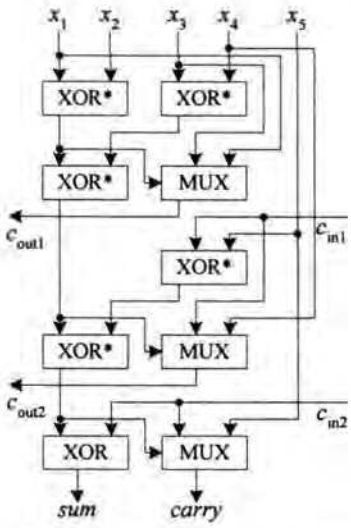


Figure 4.8 *5-2 compressor architecture - 5Δ*

In spite of the structural differences between the implementations of Fig. 4.7 and Fig. 4.8, the formulae to generate the output signals are essentially derived from the same basic architecture of Fig. 4.7. Each full adder can be logically expressed as:

$$s_{FA} = a \oplus b \oplus c \tag{4.7}$$

$$c_{FA} = (a \oplus b) \cdot c + \overline{(a \oplus b)} \cdot a \tag{4.8}$$

where a, b, and c are the primary inputs, and s_{FA} and c_{FA} are the primary outputs of the full adder.

It follows that the outputs and the internal nodes of Fig. 4.7 can be expressed by the following set of equations.

$$s_1 = x_1 \oplus x_2 \oplus x_3 \tag{4.9}$$

$$c_{out1} = (x_1 \oplus x_2) \cdot x_3 + \overline{(x_1 \oplus x_2)} \cdot x_1 \tag{4.10}$$

$$\begin{aligned} s_2 &= s_1 \oplus x_4 \oplus c_{in1} \\ &= x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus c_{in1} \end{aligned} \tag{4.11}$$

$$\begin{aligned} c_{out2} &= (x_4 \oplus s_1) \cdot c_{in1} + \overline{(x_4 \oplus s_1)} \cdot x_4 \\ &= (x_1 \oplus x_2 \oplus x_3 \oplus x_4) \cdot c_{in1} + \overline{(x_1 \oplus x_2 \oplus x_3 \oplus x_4)} \cdot x_4 \end{aligned} \tag{4.12}$$

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

$$sum = s_2 \oplus x_5 \oplus c_{in2} = x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus c_{in1} \oplus c_{in2} \quad (4.13)$$

$$carry = (x_5 \oplus s_2)\cdot c_{in2} + \overline{(x_5 \oplus s_2)}\cdot x_5 = (x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus c_{in1})\cdot c_{in2} + \overline{(x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus c_{in1})}\cdot x_5 \quad (4.14)$$

A faster 5-2 compressor proposed in [KWO00] is shown in Fig. 4.9. This architecture uses a different method to generate the outputs, c_{out1} and c_{out2}. Although this architecture produces different output bit patterns in c_{out1} and c_{out2} for the same input data, it still abides by (4.6). It is claimed to have a delay of 4Δ.

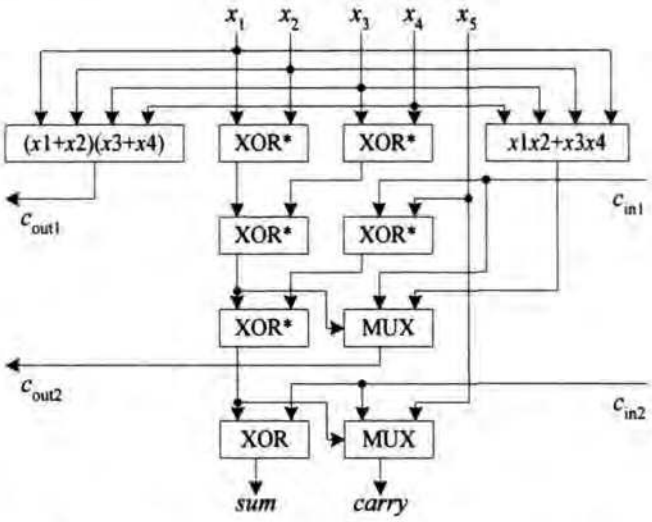


Figure 4.9 *5-2 compressor architecture - 4Δ*

(4.15) - (4.18) show the output functions:

$$sum = x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus c_{in1} \oplus c_{in2} \quad (4.15)$$

$$c_{out1} = (x_1 + x_2)\cdot(x_3 + x_4) \quad (4.16)$$

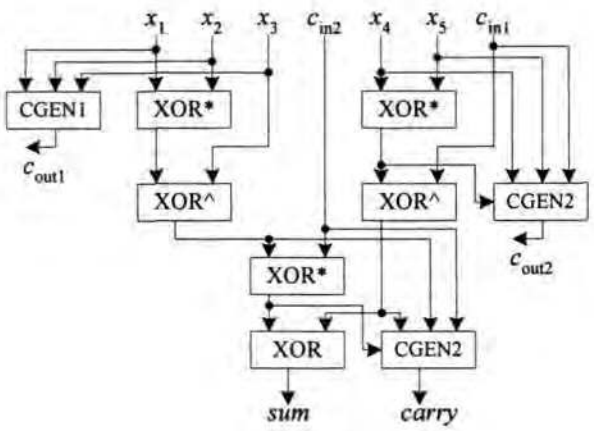
$$c_{out2} = \overline{(x_1 \oplus x_2 \oplus x_3 \oplus x_4)}\cdot(x_1 \cdot x_2 + x_3 \cdot x_4) + (x_1 \oplus x_2 \oplus x_3 \oplus x_4)\cdot c_{in1} \quad (4.17)$$

$$carry = \overline{(x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus c_{in1})}\cdot x_5 + (x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus c_{in1})\cdot c_{in2} \quad (4.18)$$

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

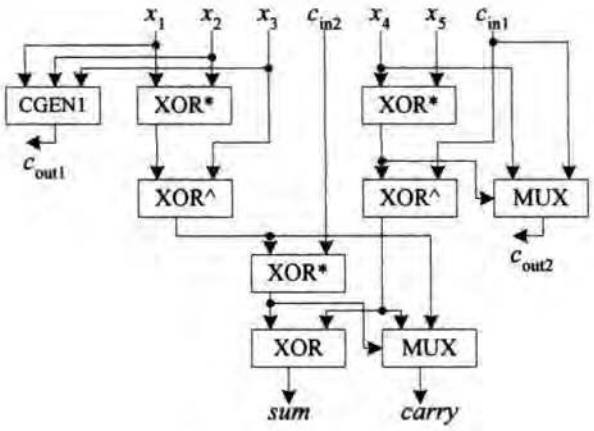
We proposed a new and faster architecture with the same theoretical critical path delay of 4Δ. The structure of our 5-2 compressor is shown in Fig. 4.10(a). It uses one less module than the 5-2 compressor of Fig. 4.9. The mechanisms for generating the carry output signals, such as *carry*, c_{out1}, c_{out2}, are fundamentally different from those of the existing architectures. The carry generator, CGEN1 is used to produce the signal c_{out1}. In anticipation that this carry generator is most likely to be situated in the longest delay path in the tree structured multiplier, it is implemented as a complex gate fed only with the primary inputs. Although several more transistors are used, the speed of this carry generator is faster than that controlled by a XOR* module. The other two carry generators CGEN2, which are used to produce the *carry* and c_{out2}, are still controlled by the corresponding XOR* modules. Unlike CGEN1, one of the inputs to CGEN2 is either c_{in1} or c_{in2}, which comes from an output of the compressor of the preceding stage. Since these signals are often arrived later than the primary inputs, implementing CGEN2 with more costly complex gate as CGEN1 is not necessary as it will not help to reduce the critical path delay of the tree structured multiplier any way.

The carry generators, CGEN2 in Fig. 4.10(a) can also be implemented by multiplexers, as shown in Fig. 4.10(b). CGEN2 of Fig. 4.10(a) is better implemented in complementary CMOS logic style, which offers low capacitive loading to the driving XOR* modules while the MUX-based carry generator modules of Fig. 4.10(b) is more suitable for realization with other logic styles.



(a)

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library



(b)

Figure 4.10 *Proposed 5-2 compressor architecture - 4Δ*

The output functions of our proposed architecture in Fig. 4.10 are described by (4.19) – (4.22)

$$sum = x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus c_{in1} \oplus c_{in2} \tag{4.19}$$

$$c_{out1} = (x_1 + x_2) \cdot x_3 + x_1 \cdot x_2 \tag{4.20}$$

$$c_{out1} = (x_4 \oplus x_5) \cdot c_{in1} + x_4 \cdot x_5 \equiv (x_4 \oplus x_5) \cdot c_{in1} + \overline{(x_4 \oplus x_5)} \cdot x_4 \tag{4.21}$$

$$\begin{aligned} carry &= \big((x_1 \oplus x_2 \oplus x_3) \oplus (x_4 \oplus x_5 \oplus c_{in1})\big) \cdot c_{in2} \\ &\quad + (x_1 \oplus x_2 \oplus x_3) \cdot (x_4 \oplus x_5 \oplus c_{in1}) \\ &\equiv \big((x_1 \oplus x_2 \oplus x_3) \oplus (x_4 \oplus x_5 \oplus c_{in1})\big) \cdot c_{in2} \\ &\quad + \overline{(x_1 \oplus x_2 \oplus x_3) \oplus (x_4 \oplus x_5 \oplus c_{in1})} \cdot (x_1 \oplus x_2 \oplus x_3) \end{aligned} \tag{4.22}$$

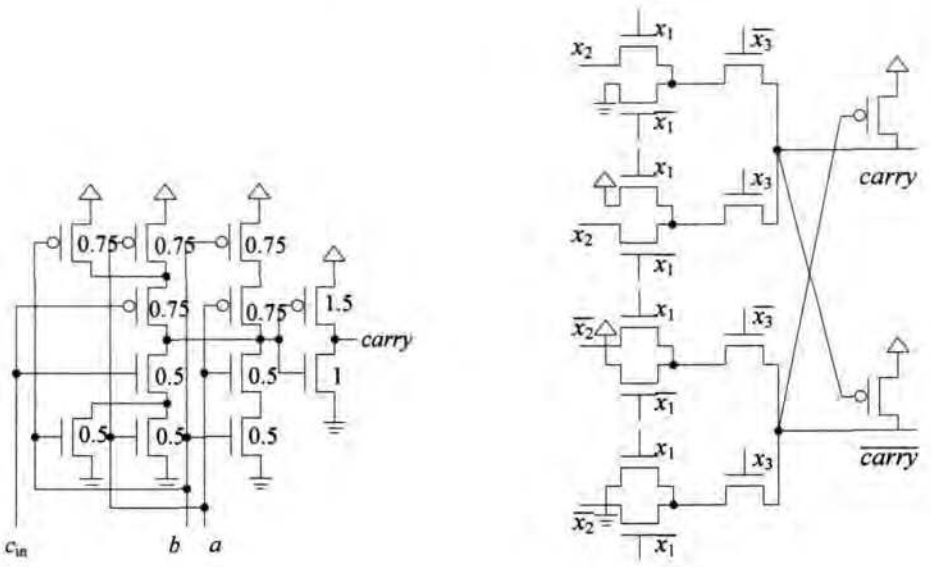
Based on the above formulae, it is conjectured that lowering the critical path delay of the 5-2 compressor to 3Δ or lower is almost impossible. However, it is very likely to explore different logic styles at transistor level to design significantly improved low power and high-speed 5-2 compressor cells for instantiation at architectural level. For example, a dual-rail 5-2 compressor [PRA01] is proposed for the architecture of Fig. 4.8, where the XOR modules are implemented as dual-rail multiplexers to improve the performance.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

4.3.2 Circuit of the proposed 5-2 compressor architectures

The design of Fig. 3.4(e) is recommended for the implementation of XOR* modules to allow low voltage operation. The XOR^ modules do not need strong driving capability for the internal modules driven by them. Therefore, the design of Fig. 3.6(a) without the output buffer can be used for the XOR^ module. The *sum* output stage is implemented with the circuit of Fig. 3.6(b) to assure the drivability. The new complementary CMOS logic style circuit of Fig. 3.8 from Chapter 3, is suggested for the implementation of the CGEN2 module.

The c_{out1} output of the CGEN1 module receives the primary inputs of x_1 to x_3 directly. It can be implemented with the carry generator circuit of the full adder in complementary CMOS logic style, which is shown in Fig. 4.11(a) [NAG96, ZIM97]. This circuit uses two more transistors than the circuit of Fig. 3.8, but it bypasses the XOR and XNOR signals from the preceding XOR* module, which has prevented the delay of the c_{out1} generation from being degraded by about 1Δ. Fig. 4.11(b) shows the circuit for the CGEN1 module implemented in CPL logic style [ZIM97].



(a) (b)

Figure 4.11 *Implementations of the CGEN1 module*

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

The complete layout of the proposed 5-2 compressor of 4Δ is shown in Fig. 4.12.

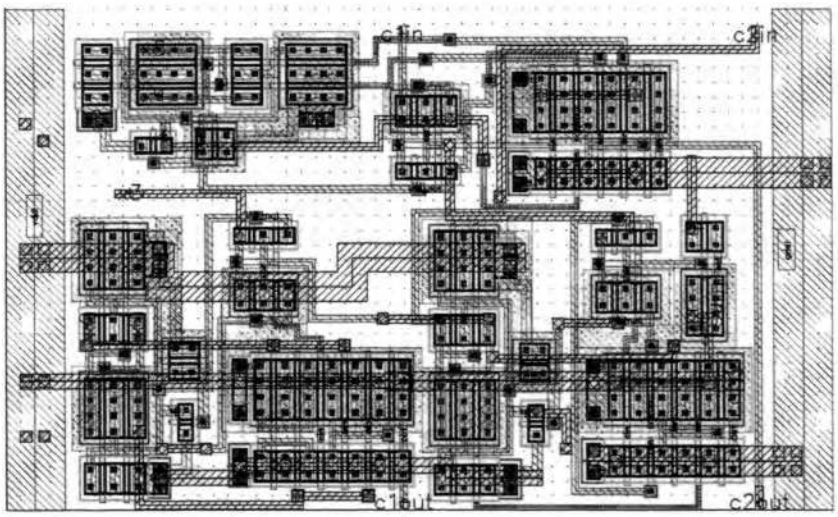


Figure 4.12 *Layout of the proposed 5-2 compressor*

4.4 Simulation results

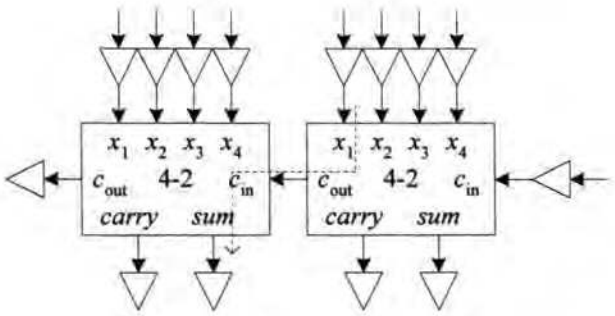
4.4.1 Simulation environment

The simulations are performed by Nassda Hsim 2.0 tool with the option "HSIMSPEED" set to "0". This option gives the slowest simulation time with the highest accuracy giving results compatible to HSPICE simulation. All the circuits and layouts are targeted for the latest Chartered Semiconductor CSM 0.18μm CMOS technology. Therefore, the circuits are designed and optimized based on this process model.

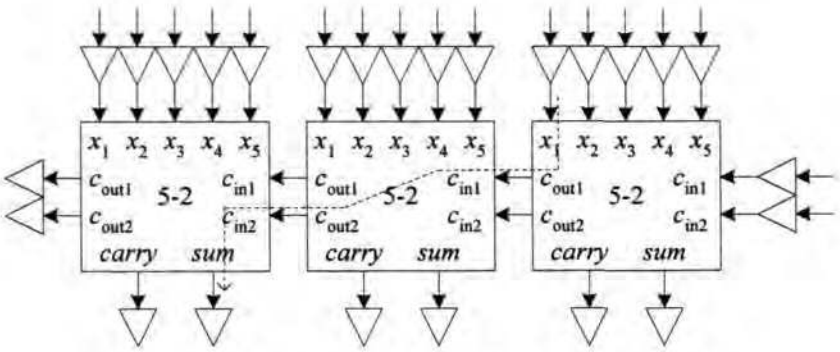
The simulation environments for the 4-2 compressor and 5-2 compressor circuits are shown in Fig. 4.13. Each input and output pins are cascaded by buffers, which provide a realistic simulation environment reflecting the compressor operation in actual applications. The simulation environments for the 4-2 compressor and 5-2 compressor consist of two cascaded 4-2 compressors and three cascaded 5-2 compressors, respectively. These compressors run in parallel to simulate an actual compressor stage in the CSA tree. More than one compressor are used in the simulation because the critical paths of some data patterns may cross adjacent compressors in the same stage of the CSA

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

tree. The dashed lines in Fig. 4.13 indicate the scenario of such potential critical paths. The leftmost compressor of both simulation environments is inspected because it is most probable to have the longest delay. The delay is measured from the earliest input signal reaching 50% of the supply voltage to the latest output signal reaching 50% of the supply voltage for each input cycle. The worst-case delay is the largest delay among all input data.



(a) *4-2 compressor simulation environment*



(b) *5-2 compressor simulation environment*

Figure 4.13 *Simulation environments*

For each circuit, 1024 data are randomly generated by MATLAB to feed into the circuits as input stimuli. The circuits under test are simulated at various supply voltages, ranging from 0.6V to 3.3V. Two simulation frequencies (the rate at which data patterns are fed) are used. When the supply voltage is larger than 1.0V, the computation rate of 100 MHz is used. When the supply voltage is equal to or lower than 1.0V, 10 MHz computation rate is used. It should be noted that the simulation frequency is not the maximum operating frequency of the compressors. In fact, the compressors simulated are capable of operating correctly at much higher frequency than the simulation frequency. The average

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

power consumption of the leftmost compressor is measured for every supply voltage, with the power consumed by the additional buffers excluded from the average power consumption calculation.

4.4.2 Simulation results of 4-2 compressors

Nine different 4-2 compressor designs are simulated. The first circuit is a full complementary CMOS logic style implementation of Fig. 4.3, which is used as the benchmark for low voltage operation. The second to the sixth compressors use the same architecture of Fig. 4.4, with the XOR modules implemented by the circuit of Fig. 3.6(b) and the MUX modules implemented by the circuit of Fig. 3.7(b) from Chapter 3. They differ mainly in the circuit implementation of the XOR* modules. The XOR* modules of these five compressors are respectively implemented by the circuits of Fig. 3.4(a) to Fig. 3.4(e) from Chapter 3. The seventh compressor is a hybrid design employing the circuit of Fig. 3.4(e) for its XOR* modules, the circuit of Fig. 3.6(b) for its XOR modules, and the circuit of Fig. 3.7(c) as its MUX modules. The eighth and ninth compressors are implemented by CPL and DPL logic styles, using the circuits of Fig. 3.7(d) and 3.4(f), and Fig. 3.7(e) and 3.4(f), respectively as their building blocks. The configurations of the nine compressors are listed in Table 4.1. For brevity, only the figure numbers of the circuits, e.g., 3.4(a) instead of Fig. 3.4(a), are shown in the table. The last column in Table 4.1 shows the lowest operable voltage for each 4-2 compressor obtained from the simulation, below which the circuit fails to function correctly. The full simulation results for the performance of all the compressors at different supply voltages are tabulated in Table 4.2 to Table 4.4. Two best performances at each supply voltage are highlighted in italic and bold print for ease of comparison.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Table 4.1 *Configurations of the simulated 4-2 compressors*

S/N	Design	XOR-XNOR	XOR	MUX	Min. V_{DD} (V)
1	4-2_CMOS	4.3			0.6
2	4-2_a	3.4(a)	3.6(b)	3.7(b)	0.9
3	4-2_b	3.4(b)	3.6(b)	3.7(b)	0.6
4	4-2_c	3.4(c)	3.6(b)	3.7(b)	2.1
5	4-2_d	3.4(d)	3.6(b)	3.7(b)	0.8
6	4-2_e	3.4(e)	3.6(b)	3.7(b)	0.6
7	4-2_hybrid	3.4(e)	3.6(b)	3.7(c)	0.6
8	4-2_CPL	3.7(d) + 3.4(f)	3.7(d) + 3.4(f)	3.7(d)	0.6
9	4-2_DPL	3.7(e) + 3.4(f)	3.7(e) + 3.4(f)	3.7(e)	0.6

Table 4.2 *Comparison of delay* (ns) *of 4-2 compressors*

Delay (ns)	0.6V	0.7V	0.8V	0.9V	1.0V	1.2V	1.5V	1.8V	2.1V	2.5V	3.3V
4-2_CMOS	***6.289***	***3.184***	2.051	1.484	1.16	0.811	0.574	0.457	0.394	0.36	0.31
4-2_a				12.69	5.989	1.912	0.745	0.458	***0.36***	***0.3***	***0.26***
4-2_b	8.032	4.05	2.538	1.798	1.364	0.898	0.59	0.46	0.401	0.357	0.305
4-2_c									1.01	0.75	0.52
4-2_d			34.66	7.043	3.696	1.54	0.789	0.548	0.432	0.352	0.33
4-2_e	6.825	3.43	2.2	1.56	1.19	0.804	0.554	0.455	0.403	0.365	0.326
4-2_hybrid	7.11	3.724	2.425	1.736	1.357	0.934	0.656	0.54	0.47	0.42	0.35
4-2_CPL	***6.28***	***3.04***	***1.913***	***1.375***	***1.07***	***0.74***	***0.53***	***0.44***	0.39	0.35	0.32
4-2_DPL	6.735	3.323	***2.04***	***1.402***	***1.04***	***0.69***	***0.48***	***0.38***	***0.33***	***0.29***	***0.255***

Table 4.3 *Comparison of power* (μW) *of 4-2 compressors*

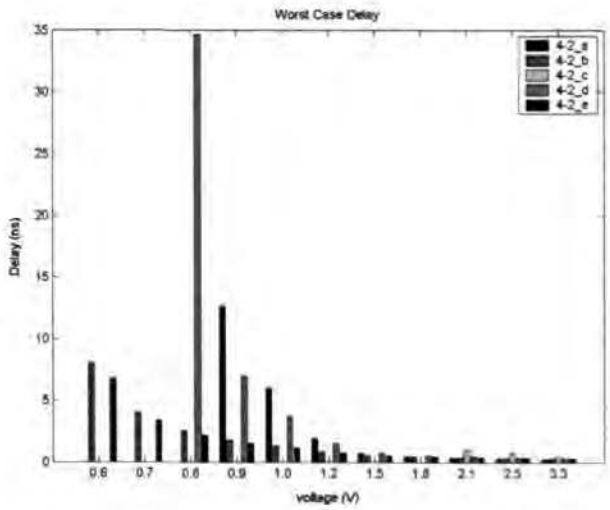
Power (μW)	0.6V	0.7V	0.8V	0.9V	1.0V	1.2V	1.5V	1.8V	2.1V	2.5V	3.3V
4-2_CMOS	0.1669	0.2327	0.3161	0.4049	0.4801	0.7601	1.332	2.101	3.228	5.025	9.8
4-2_a				0.3299	0.409	0.6763	1.326	2.67	5.259	12.73	70.66
4-2_b	0.1142	0.1573	0.2147	0.2698	0.333	0.5098	0.9125	1.496	2.38	3.875	8.23
4-2_c									14.07	28.28	98.42
4-2_d			0.2614	0.3158	0.3681	0.5424	0.8916	1.325	1.984	3.002	***6.028***
4-2_e	***0.1123***	***0.1529***	***0.2039***	***0.264***	***0.3102***	***0.4781***	***0.8193***	***1.258***	***1.885***	***2.944***	***6.023***
4-2_hybrid	***0.1137***	***0.1552***	0.2074	0.2649	0.3169	0.4936	0.8413	***1.275***	***1.93***	***2.998***	6.143
4-2_CPL	0.1266	0.1778	0.2323	0.2944	0.3565	0.5657	0.9617	1.497	2.338	3.617	7.863
4-2_DPL	0.1186	0.1606	***0.2068***	***0.2526***	***0.3058***	***0.4839***	***0.8322***	1.301	2.031	3.461	8.409

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Table 4.4 *Comparison of power efficiency* (fJ) *of 4-2 compressors*

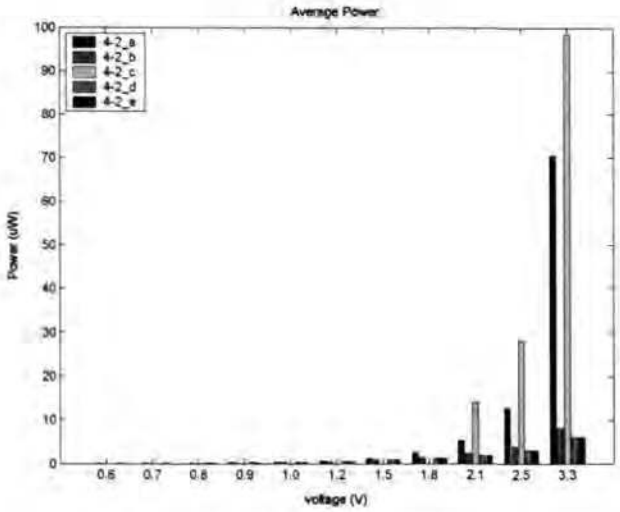
PDP (fJ)	0.6V	0.7V	0.8V	0.9V	1.0V	1.2V	1.5V	1.8V	2.1V	2.5V	3.3V
4-2_CMOS	1.050	0.7409	0.6483	0.6009	0.5569	0.6164	0.7646	0.9602	1.272	1.809	3.038
4-2_a				4.186	2.450	1.293	0.9879	1.223	1.893	3.819	18.37
4-2_b	0.9173	0.6371	0.5449	0.4851	0.4542	0.4578	0.5384	0.6882	0.9544	1.383	2.510
4-2_c									14.21	21.21	51.18
4-2_d			9.060	2.224	1.360	0.8353	0.7035	0.7261	0.8571	***1.057***	***1.989***
4-2_e	***0.7664***	***0.5244***	0.4486	0.4118	***0.3691***	***0.3844***	***0.4539***	***0.5724***	***0.7597***	1.075	***1.963***
4-2_hybrid	0.8084	0.5780	0.5029	0.4599	0.4300	0.4610	0.5519	0.6885	0.9071	1.259	2.150
4-2_CPL	***0.7950***	0.5405	***0.4444***	***0.4048***	0.3815	0.4186	0.5097	0.6587	0.9118	1.266	2.516
4-2_DPL	0.7988	***0.5337***	***0.4219***	***0.3541***	***0.3180***	***0.3339***	***0.3995***	***0.4944***	***0.6702***	***1.004***	2.144

To investigate the performance variations of the same 4-2 architecture due to different implementations of the XOR-XNOR module, the worst case delay, power dissipation and power-delay product of Designs 2 – 6 are charted in Fig. 4.14. It is evident that the worst-case delays at low supply voltages of Designs 2 and 5 are much longer than those of the other designs. Designs 2 and 4 consume more power. Consequently, Designs 3 and 6 perform significantly better than the other designs in terms of the power efficiency. In fact, these are the only two designs of Fig. 4.14 that are able to operate down to 0.6V. The circuits used to implement the XOR* modules for Designs 3 and 6 are respectively, Fig. 3.4(b) and our proposed Fig. 3.4(e).

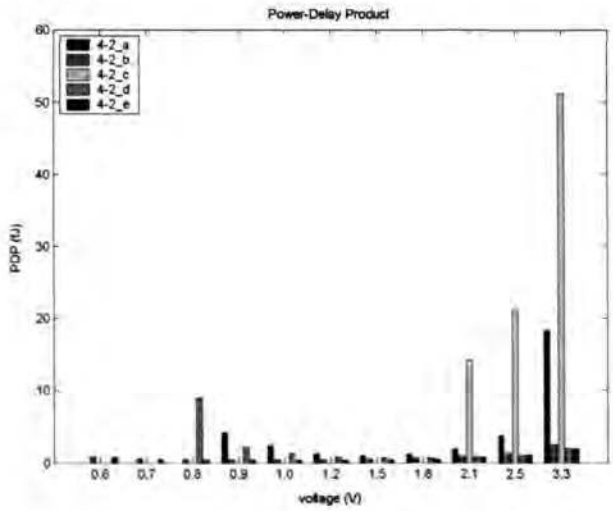


(a) *Worst case delay*

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library



(b) *Average power*



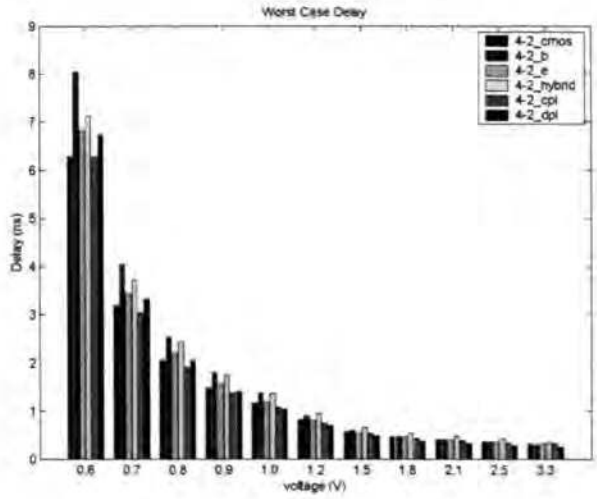
(c) *Power-delay product*

Figure 4.14 *Performances of 4-2 compressors (Designs 2 – 6)*

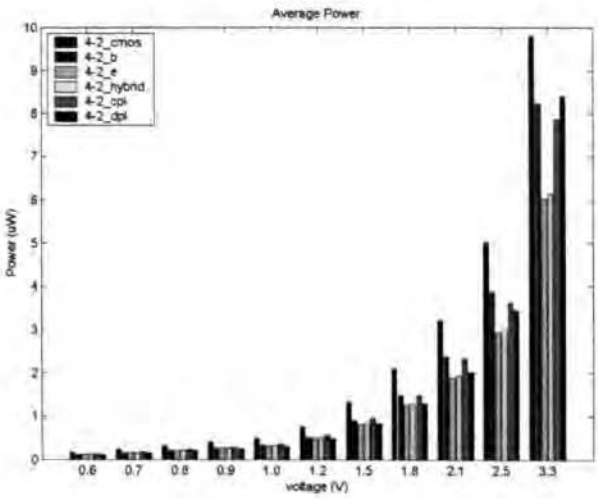
The worst case delay, power dissipation and power-delay product of all designs capable of functioning down to the lowest supply voltage of 0.6V, including our proposed hybrid 4-2 compressor are charted in Fig. 4.15. Designs 1, 8 and 9 featuring respectively the CMOS, CPL and DPL logic styles have comparatively shorter worst-case delay. However, they and Design 3 dissipate notably higher power than Designs 6 and 7, which use our proposed XOR* cell of Fig. 3.4(e). For example, Design 8 (CPL) consumes 12.7% more power at 0.6V than Design 6 (4-2_e). The average power consumption

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

exacerbates with increasing supply voltages to a surplus of 14.9% at 1.0V, 19.0% at 1.8V and 30.5% at 3.3V. The power efficiency (PDP) of Designs 6 and 7 are comparable to those circuits implemented in CPL and DPL styles. Both CPL and DPL logic styles require the generation of dual-rail signals for each primary input and output, incurring almost twice as many interconnecting lines as the other designs. Taking into account the substantial capacitive load due to the wiring overhead, Designs 6 and 7 will not be competitive for the implementation of 4-2 compressors in large parallel multipliers.

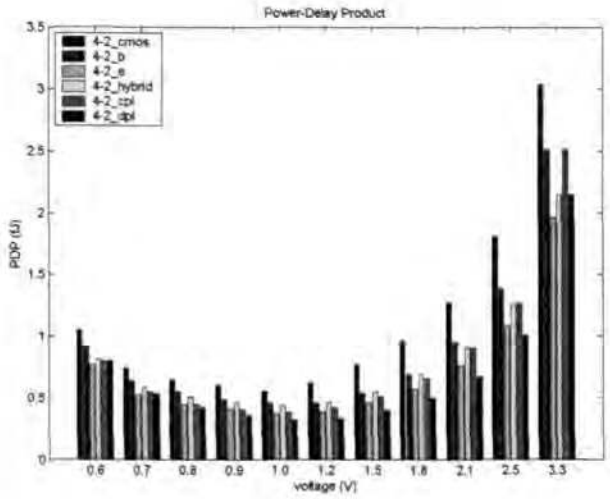


(a) *Worst case delay*



(b) *Average power*

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library



(c) *Power-delay product*

Figure 4.15 *Performances of 4-2 compressors (Designs 1, 3, 6 – 9)*

4.4.3 Simulation Results of 5-2 Compressors

Thirteen different 5-2 compressor designs are simulated. These selected designs are all operable from 0.6V to 3.3V and their circuit configurations are tabulated in Table 4.5. Each design is named according to its base architecture and a postfix indicating the types of circuits employed for the three constituent modules. For example, the name 5del_ebb of Design 2 implies that it is a 5Δ 5-2 compressor constructed by the circuits of Fig. 3.4(e), Fig. 3.6(b) and Fig. 3.7(b) from Chapter 3 for its XOR*, XOR and MUX modules, respectively. The first five designs are 5-2 compressors of 5Δ based on the architecture of Fig. 4.8. The four designs that followed are the 5-2 compressors of 4Δ based on the architecture of Fig. 4.9 proposed by Kwon *et. al.* [KWO00]. Designs 6 – 8 use the optimized complementary CMOS logic style to generate the OR-AND and AND-OR functions while Design 9 (kwon_cpl) generates these two functions with CPL logic style. The last four designs are the 5-2 compressors of 4Δ using the proposed architecture of Fig. 4.10. Designs 10, 11 and 13 use the circuits of Fig. 4.10(b) and Fig. 4.11(a) for the CGEN1 modules, and the circuit of Fig. 4.7(b) for the MUX modules. Design 12 is based on the architecture of Fig. 4.10(a), with a hybrid composition of optimally designed circuits of different logic styles. It uses the proposed XOR* cell of Fig. 3.4(e), the pass-

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

transistor style XOR gate of Fig. 3.6(b), the XOR^ circuit of Fig. 3.6(a), the complementary CMOS styled CGEN1 circuit of Fig. 4.11(a), and the CGEN2 circuit of Fig. 3.8. Design 13 (CPL) is based on the same architecture as Designs 10 and 11, but uses the circuits implemented in CPL logic style for the XOR*, XOR, XOR^ and MUX modules. Its CGEN1 module is implemented with the CPL circuit of Fig. 4.11(b).

Table 4.5 *Configurations of the simulated 5-2 compressors*

S/N	Design	Architecture	XOR*	XOR, XOR^	MUX, CGEN
1	**5del_bbb**	4.8	3.4(b)	3.6(b)	3.7(b)
2	**5del_ebb**	4.8	3.4(e)	3.6(b)	3.7(b)
3	**5del_hybrid**	4.8	3.4(e)	3.6(b)	3.7(c)
4	**5del_cpl**	4.8	3.7(d) + 3.4(f)	3.7(d) + 3.4(f)	3.7(d)
5	**5del_dpl**	4.8	3.7(e) + 3.4(f)	3.7(e) + 3.4(f)	3.7(e)
6	**kwon_bbb**	4.9	3.4(b)	3.6(b)	3.7(b)
7	**kwon_ebb**	4.9	3.4(e)	3.6(b)	3.7(b)
8	**kwon_ebc**	4.9	3.4(e)	3.6(b)	3.7(c)
9	**kwon_cpl**	4.9	3.7(d) + 3.4(f)	3.7(d) + 3.4(f)	3.7(d)
10	**4del_bbb**	4.10(b)	3.4(b)	3.6(b), 3.6(a)	4.11(a), 3.7(b)
11	**4del_ebb**	4.10(b)	3.4(e)	3.6(b), 3.6(a)	4.11(a), 3.7(b)
12	**4del_hybrid**	4.10(a)	3.4(e)	3.6(b), 3.6(a)	4.11(a), 3.8
13	**4del_cpl**	4.10(b)	3.7(d) + 3.4(f)	3.7(d) + 3.4(f)	4.11(b), 3.7(d)

The simulation results of the delay, power and power efficiency of all the compressors are tabulated in Table 4.6 to Table 4.8. Two best performances at each supply voltage are printed in bold and italic. The performances of all 5-2 compressors based on the 5Δ architecture are charted in Fig. 4.16 for comparison. The CPL (5del_cpl) and DPL (5del_dpl) designs have the best worst-case delay and power efficiency. Among the non dual-rail designs, 5del_bbb consumes more power and 5del_hybrid has the longest delay. Design 5del_ebb provides the best trade-off between delay, power and power efficiency among the three non dual-rail compressors. Considering all aspects of the performance, this architecture is best implemented with either CPL or DPL circuits.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Table 4.6 *Comparison of delay* (ns) *of 5-2 compressors*

Delay (ns)	0.6V	0.7V	0.8V	0.9V	1.0V	1.2V	1.5V	1.8V	2.1V	2.5V	3.3V
5del_bbb	11.42	5.722	3.583	2.52	1.909	1.262	0.839	0.68	0.588	0.518	0.45
5del_ebb	11.73	5.76	3.59	2.54	1.94	1.302	0.913	0.747	0.663	0.607	0.568
5del_hybrid	12.15	6.192	3.889	2.749	2.114	1.443	0.998	0.796	0.685	0.607	0.549
5del_cpl	***8.11***	***3.99***	***2.51***	***1.83***	***1.42***	***0.99***	***0.69***	***0.57***	***0.49***	***0.45***	0.41
5del_dpl	8.97	***4.35***	***2.65***	***1.83***	***1.371***	***0.92***	***0.64***	***0.52***	***0.45***	***0.39***	***0.35***
kwon_bbb	10.35	5.32	3.37	2.36	1.78	1.13	0.7978	0.6554	0.5668	0.4972	0.413
kwon_ebb	11.57	5.827	3.47	2.47	1.9	1.29	0.9087	0.7412	0.651	0.598	0.559
kwon_ebc	11.68	5.905	3.66	2.588	1.987	1.349	0.945	0.768	0.673	0.607	0.553
kwon_cpl	***8.92***	4.351	2.749	1.974	1.537	1.094	0.8109	0.6733	0.592	0.5437	0.4957
4del_bbb	12.33	6.11	3.78	2.63	2.01	1.34	0.91	0.73	0.62	0.55	0.46
4del_ebb	10.31	5.26	3.414	2.436	1.842	1.183	0.79	0.62	0.53	0.46	***0.39***
4del_hybrid	10.11	5.24	3.36	2.41	1.86	1.27	0.87	0.695	0.592	0.511	0.425
4del_cpl	10.14	4.66	2.92	2.07	1.62	1.11	0.8	0.66	0.57	0.51	0.47

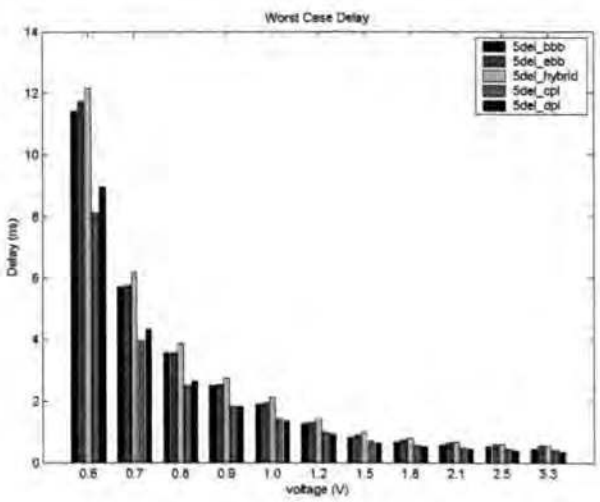
Table 4.7 *Comparison of power* (μW) *of 5-2 compressors*

Power (μW)	0.6V	0.7V	0.8V	0.9V	1.0V	1.2V	1.5V	1.8V	2.1V	2.5V	3.3V
5del_bbb	0.215	0.2947	0.398	0.51	0.6155	0.9545	1.706	2.734	4.247	6.932	15.15
5del_ebb	0.2102	0.2843	0.3797	0.4815	0.5752	0.8836	1.497	2.317	3.451	5.413	11.22
5del_hybrid	0.2128	0.2901	0.3848	0.4942	0.5938	0.9193	1.546	2.385	3.539	5.475	11.39
5del_cpl	0.2164	0.3015	0.3942	0.4974	0.596	0.9258	1.554	2.415	3.642	5.788	12.23
5del_dpl	0.1952	0.266	0.3396	0.4117	0.5007	0.7817	1.315	2.065	3.194	5.324	12.33
kwon_bbb	0.2106	0.2929	0.3929	0.5026	0.6037	0.9628	1.666	2.711	4.161	6.932	14.88
kwon_ebb	0.2094	0.2855	0.3817	0.4866	0.586	0.9048	1.519	2.4	3.66	5.827	12.21
kwon_ebc	0.2079	0.2844	0.3803	0.4852	0.584	0.91	1.539	2.44	3.65	5.788	12.12
kwon_cpl	0.2629	0.3667	0.4742	0.6046	0.7184	1.14	1.873	2.922	4.429	7.134	15
4del_bbb	0.1684	0.234	0.314	0.4039	0.4865	0.7566	1.332	2.132	3.301	5.363	11.33
4del_ebb	***0.1549***	***0.2105***	***0.2845***	***0.3598***	***0.4291***	***0.6767***	***1.137***	***1.781***	***2.695***	***4.353***	***9.02***
4del_hybrid	***0.151***	***0.207***	***0.2805***	***0.3561***	***0.4271***	***0.6696***	***1.142***	***1.757***	***2.692***	***4.308***	***9.05***
4del_cpl	0.2262	0.3158	0.4161	0.5268	0.6578	0.9911	1.701	2.684	4.031	6.338	13.67

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

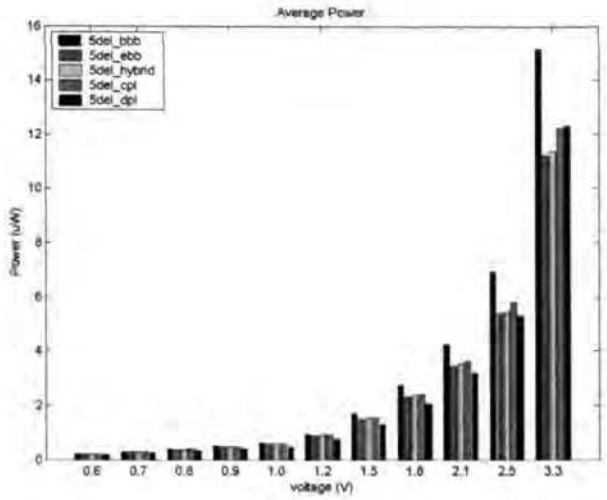
Table 4.8 *Comparison of power efficiency* (fJ) *of 5-2 compressors*

PDP (fJ)	0.6V	0.7V	0.8V	0.9V	1.0V	1.2V	1.5V	1.8V	2.1V	2.5V	3.3V
5del_bbb	2.456	1.686	1.426	1.285	1.1750	1.205	1.431	1.859	2.497	3.591	6.818
5del_ebb	2.466	1.638	1.363	1.223	1.116	1.150	1.367	1.731	2.288	3.286	6.373
5del_hybrid	2.586	1.796	1.496	1.359	1.255	1.327	1.543	1.898	2.424	3.323	6.253
5del_cpl	1.755	1.203	0.9894	0.9102	0.8463	0.9165	1.072	1.377	1.785	2.605	5.014
5del_dpl	1.751	1.157	0.8999	***0.7534***	***0.6865***	***0.7192***	***0.8416***	***1.074***	***1.437***	***2.076***	4.316
kwon_bbb	2.179	1.558	1.324	1.186	1.075	1.088	1.329	1.777	2.358	3.447	6.145
kwon_ebb	2.423	1.664	1.324	1.202	1.113	1.167	1.380	1.779	2.383	3.485	6.825
kwon_ebc	2.429	1.679	1.392	1.256	1.160	1.228	1.454	1.874	2.456	3.513	6.702
kwon_cpl	2.345	1.596	1.304	1.193	1.104	1.248	1.519	1.967	2.622	3.879	7.436
4del_bbb	2.076	1.430	1.187	1.062	0.9779	1.014	1.212	1.556	2.047	2.950	5.212
4del_ebb	***1.597***	***1.107***	***0.9713***	0.8765	***0.7904***	***0.8005***	***0.8982***	***1.104***	***1.428***	***2.002***	***3.518***
4del_hybrid	***1.527***	***1.085***	***0.9425***	***0.8583***	0.7945	0.8504	0.9935	1.221	1.594	2.201	***3.846***
4del_cpl	2.293	1.472	1.215	1.090	1.066	1.100	1.361	1.771	2.298	3.232	6.425

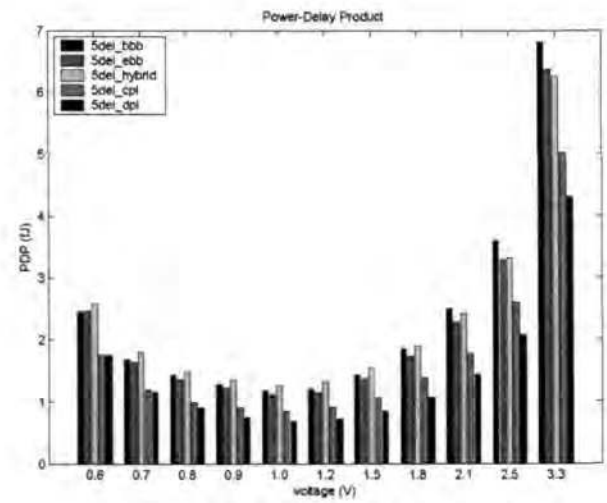


(a) *Worst case delay*

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library



(b) *Average power*



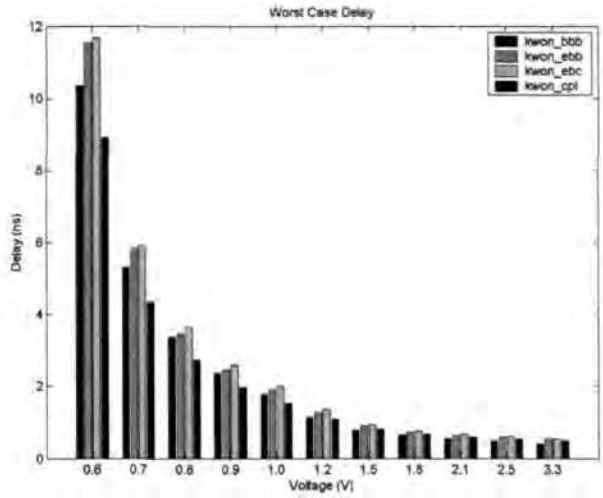
(c) *Power-delay product*

Figure 4.16 *Performances of 5-2 compressors (Designs 1 – 5)*

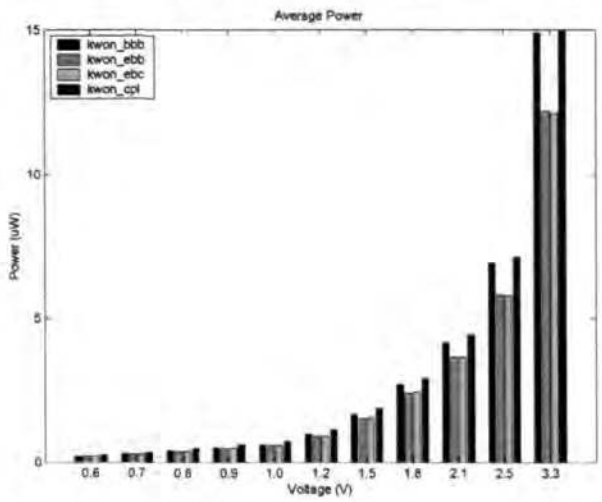
Fig. 4.17 compares the performances of various 4Δ 5-2 compressors built upon the architecture proposed by Kwon et. al. [KWO00]. The design implemented with CPL circuits continues to perform well in delay than the non dual-rail designs as in the previous comparison but its overall performance is not necessarily better than the other designs this time. The problem lies in its high power dissipation. Designs kwon_ebb and kwon_ebc consume about 2% to 18% lesser power than the other two designs, but they

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

are slower too. In terms of power efficiency, Design kwon_bbb outperforms the other two non-CPL designs, which have very similar performance.

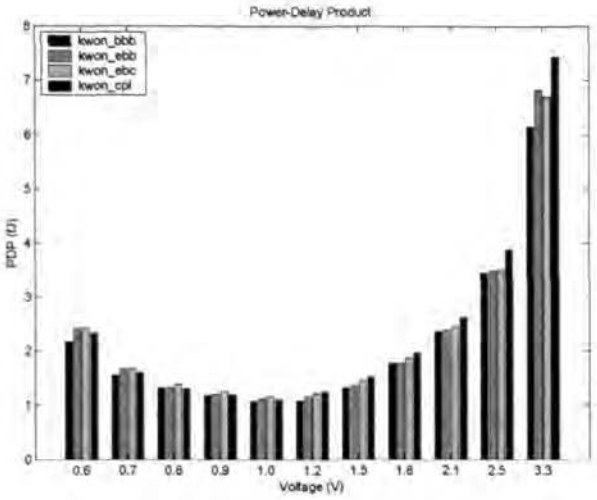


(a) *Worst case delay*



(b) *Average power*

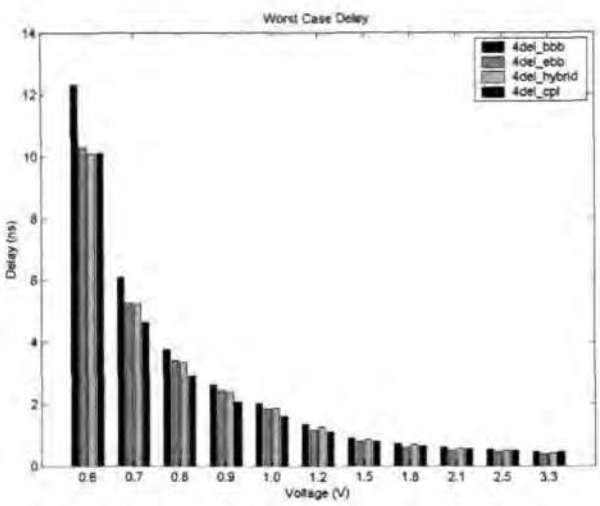
ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library



(c) *Power-delay product*

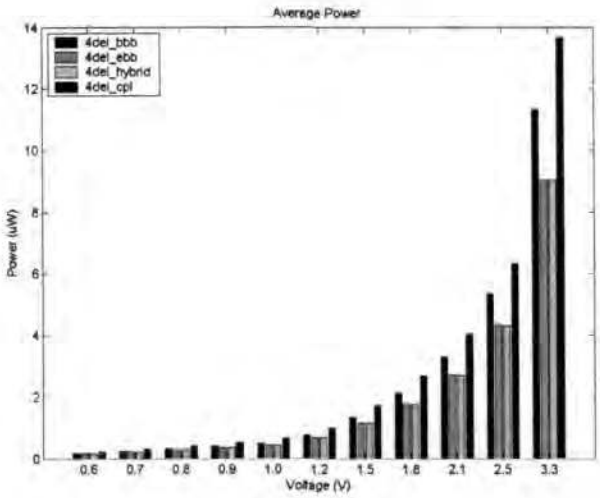
Figure 4.17 *Performances of 5-2 compressors (Designs 6 – 9)*

Fig. 4.18 shows the performances of several circuits built around our proposed 4Δ 5-2 compressor architecture. Both 4del_ebb and 4del_hybrid outperform 4del_bbb and 4del_cpl remarkably. Their speeds are comparable to the circuit implemented with CPL logic style. They consume 32% to 35% lesser power than 4del_cpl and 10% to 20% lesser power than 4del_bbb. At voltage higher than 1.0V, 4del_ebb has a slight edge over 4del_hybrid, whereas at voltage lower than 1.0V, 4del_hybrid performs slightly better.

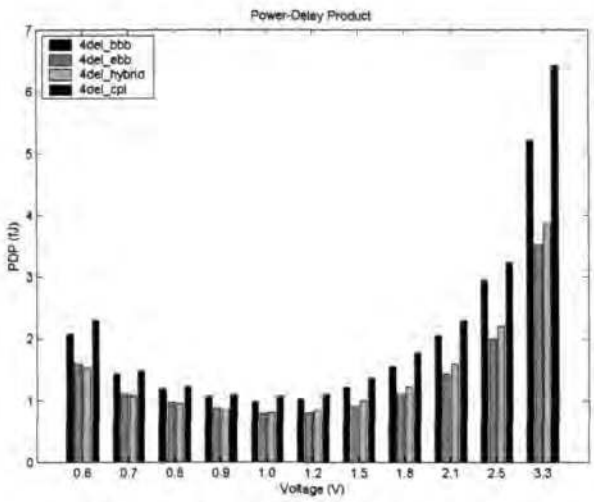


(a) *Worst case delay*

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library



(b) *Average power*



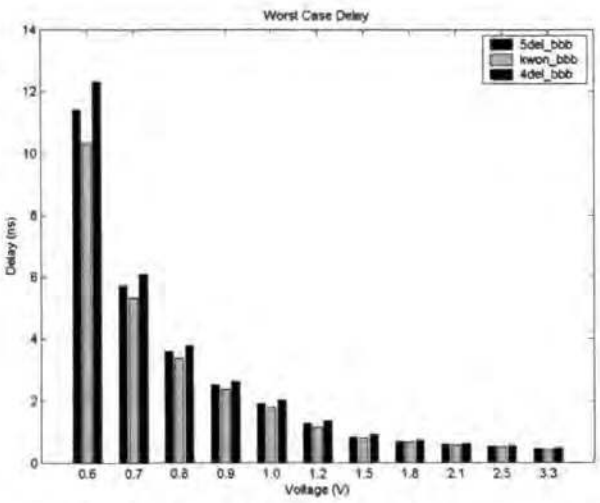
(c) *Power-delay product*

Figure 4.18 *Performances of 5-2 compressors (Designs 10 – 13)*

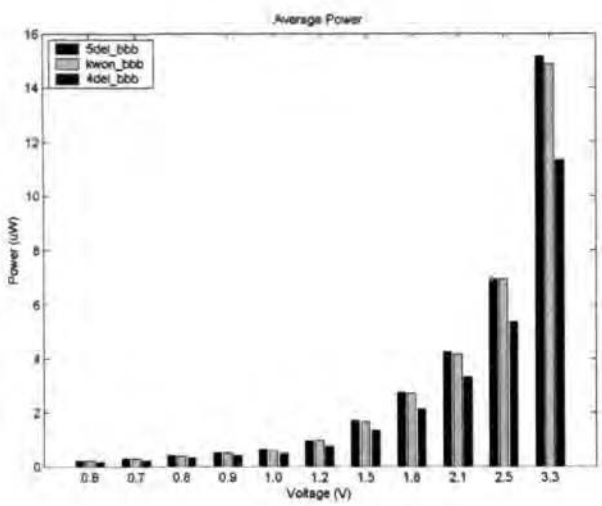
Performance differences due to architectural difference are also studied by comparing different compressor architectures using identical configuration for the same anatomized modules. The performances of three 5-2 compressor architectures with configurations of bbb, ebb, hybrid and CPL are respectively charted in Fig. 4.19 – 4.22. Design 5del_dpl is also added to Fig. 4.22 for comparison. Our proposed architecture has the best performance among all architectures implemented with non dual-rail logic styles. For modules implemented with CPL or DPL logic style, the architecture of Fig. 4.8 is better.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Fig. 4.19 compares the architectures with bbb configuration. The results show that the average power of our proposed 4Δ architecture is 19% to 24% lesser than Kwon's architecture of Fig. 4.9, and 20% to 25% lesser than the 5Δ architecture of Fig. 4.8. Although it is slower, its power-delay product is still far lower than the other architectures.

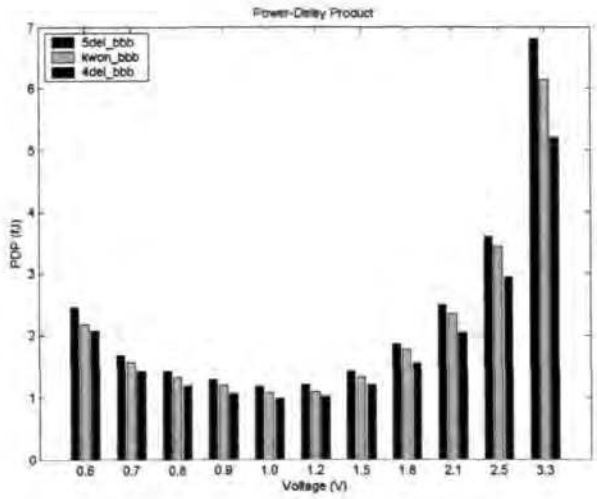


(a) *Worst case delay*



(b) *Average power*

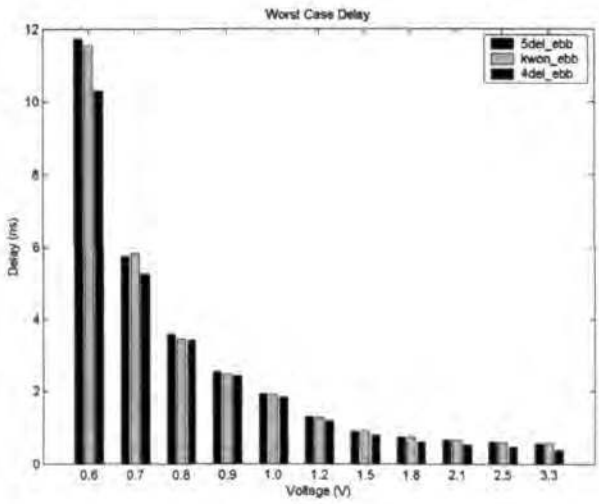
ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library



(c) *Power-delay product*

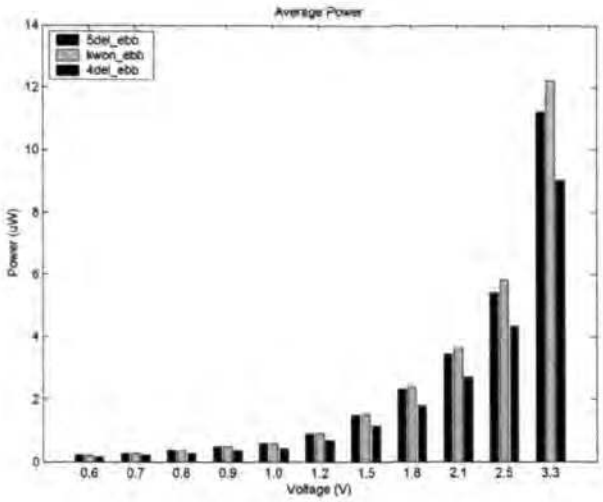
Figure 4.19 *Performances of different 5-2 compressor architectures with bbb configuration (Designs 1, 6, 10)*

Fig. 4.20 shows that, with the configuration ebb, both the average power and worst-case delay of our proposed 4Δ architecture are superior to Kwon's architecture and the 5Δ architecture. As a result, its power efficiency performance is 27% to 48% better than Kwon's and 28% to 45% better than the 5Δ architecture.

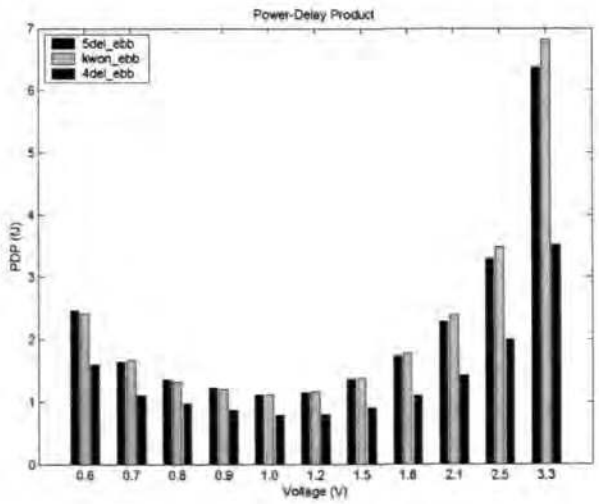


(a) *Worst case delay*

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library



(b) *Average power*

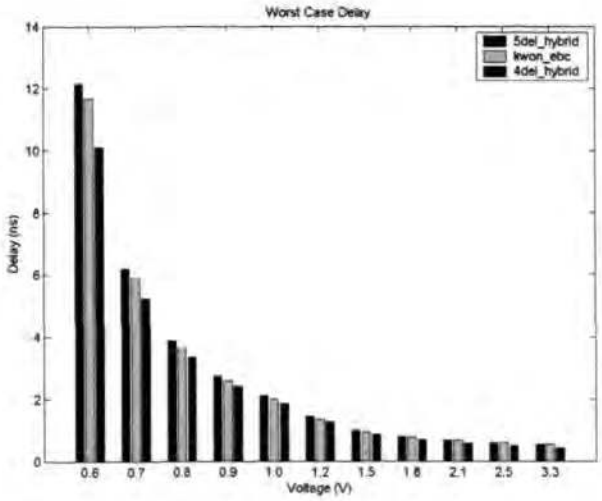


(c) *Power-delay product*

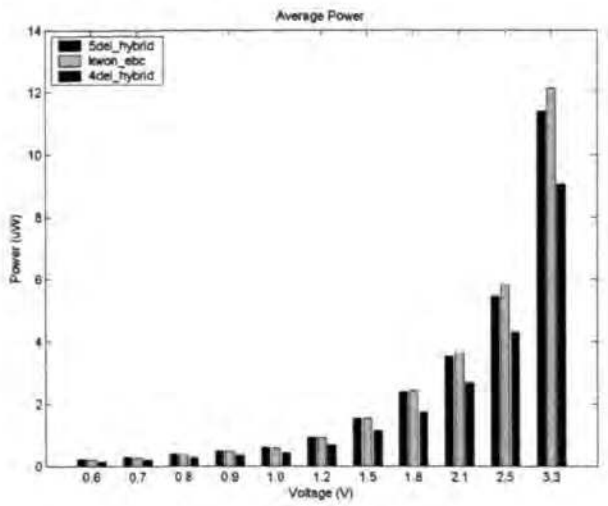
Figure 4.20 *Performances of different 5-2 compressor architectures with ebb configuration (Designs 2, 7, 11)*

Fig. 4.21 shows that, with the hybrid configuration, our proposed 4Δ architecture consumes 25% to 28% lesser power than Kwon's architecture, and 20% to 29% lesser power than the 5Δ architecture. It is also 6% to 23% faster than Kwon's architecture and 12% to 23% faster than the 5Δ architecture. Therefore, the hybrid configuration is best suited for the proposed architecture.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

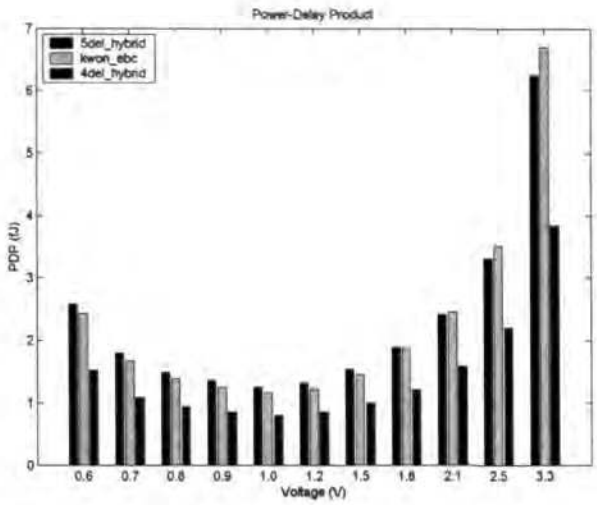


(a) *Worst case delay*



(b) *Average power*

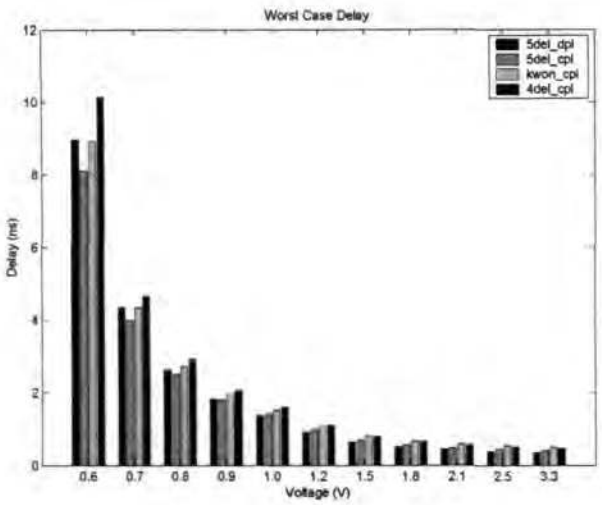
ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library



(c) *Power-delay product*

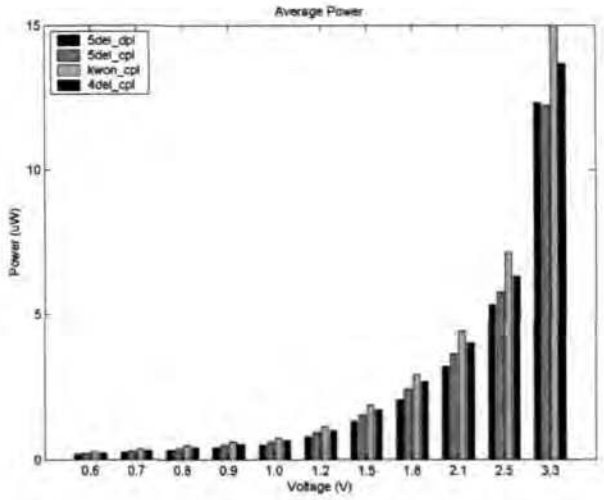
Figure 4.21 *Performances of different 5-2 compressor architectures with hybrid configuration (Designs 3, 8, 12)*

Fig. 4.22 shows the comparison of the architectures with CPL and DPL configurations. The designs of the 5Δ architectures, 5del_cpl and 5del_dpl, have their performances improved sensibly over the other architectures. Therefore, the CPL and DPL logic styles are more suitable to be implemented on this architecture.

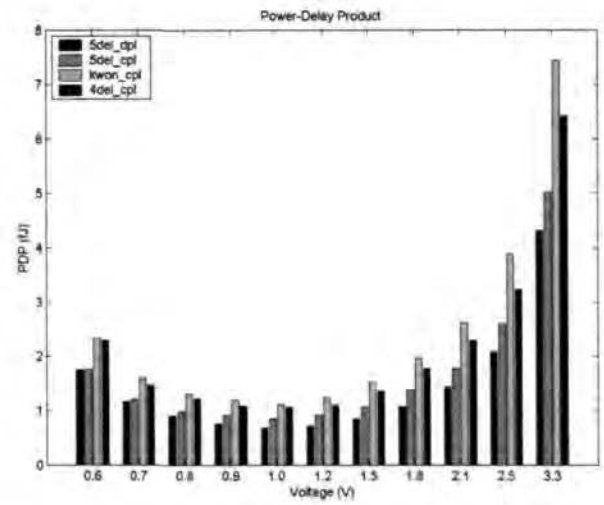


(a) *Worst case delay*

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library



(b) *Average power*



(c) *Power-delay product*

Figure 4.22 *Performances of different 5-2 compressor architectures of dual-rail logic configuration (Designs 4, 5, 9, 13)*

4.5 Summary

The architectures of 4-2 and 5-2 compressors are analyzed and different CMOS logic style circuit implementations of their constituent modules are explored. A novel 5-2 compressor architecture of 4Δ delay is also proposed. In order to realistically assess and compare the figures of merits of different configurations of 4-2 and 5-2 compressors at

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

various supply voltages, new simulation environments are established to ensure the measured performances are still sustainable when these cells are integrated in a CSA tree. The simulation results show that the 4-2 and 5-2 compressors constructed with the novel XOR* cell is able to function down to 0.6V, and features high speed and low power characteristics. Our proposed 5-2 compressor architecture outperforms all the other architectures over the range of voltages simulated, particularly when it is configured with the proposed circuits for the XOR* and the carry generator modules. Better performances against other architectures are also attained almost irrespective of the logic styles used for the circuit implementation of their constituent modules. In summary, a library of excellent power efficiency 4-2 and 5-2 compressor cells based on CMOS process technology has been developed for implementing high speed and low power multipliers operable at ultra low supply voltages.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Chapter 5

An Area and Energy Efficient IP Core for Scalar Product Computation

5.1 Introduction

Contemporary digital signal processing algorithms for image processing and telecommunication applications are increasingly predominated by matrix- and vector-like arithmetic where the same kind of simple or compound operations are carried out on different elements for two or more vectors [NAY99, OKA91, PAR00]. Examples of algorithms and applications that require such computations in their implementations are the discrete Fourier transform, the discrete cosine transform and wavelet transform used widely in speech and image processing applications [GRG01, NAY99, TON95, YU02], and finite impulse response, infinite impulse response digital filtering and adaptive decision-directed least mean square algorithms used in signal identification, waveform shaping, channel equalization, and magnetic storage technology [HAS01, MUH01, WON95, SHA93, GUN99]. Scalar (or inner) product computation forms the basis of such special purpose complex arithmetic that is inefficiently handled by software on the core central processing unit or general-purpose digital signal processor. Therefore, hardware-based implementation of scalar product computations [BRE98, DAD87, LIN01] is not only desirable, but also indispensable if we are to achieve the processing capacity required by applications such as real-time digital video and mobile communication systems Several widely used examples of which are digital cameras, mobile phone, PDAs and pocket PCs. Low voltage operation and low power dissipation are required to perform these computation functions since the power supply of the mobile devices is

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

often limited from 1 to several battery cells. The trend of wireless communication system has been to move the analog-to-digital converter closer to the antenna to perform more functions digitally at an increasingly higher data rate. Therefore, the processing speed and power requirements of the digital modem and channel equalization functions will not be achievable without some form of a dedicated ASIC preprocessor [WON95, SHA93, GUN99]. Furthermore, the development of such a preprocessor or macrocell has been deterred by the excessive use of silicon area and numerous interconnections and interminable layout details to be managed and optimized manually. Because of the restricted VLSI area of the system-on-chip (SoC) applications, the design of such macrocells has often been restricted to small word widths and limited input vector sizes.

In this chapter, we present a new scalar product macrocell focusing on deriving the benefits of power, delay and area reduction efficiency for emerging process technology from the inception of algorithmic changes to architectural and physical design space explorations. Supported by the on chip auxiliary circuits, the macrocell is able to be configured either as a normal vector multiplier or as a digital filter, thus making itself versatile in a number of applications. The proposed algorithm has led to the creation of a low power, high performance bit parallel architecture for the computation of scalar products. While the number of transistors required to implement a macrocell with large input vector sizes and operand width is inevitably high, it is mitigated by a significant increase in VLSI area usage density in our method. Our architecture possesses high modularity and hierarchical regularity, enabling the use of short local interconnections and thus simplifying the global interconnects. Unlike the conventional architecture, the signal arrival time at the inputs of each module is much easier to equalize to reduce glitches and spurious transitions. Thanks to the hierarchical structure, the amount of nonrecurring engineering work is small as only one-time customization of one partial product accumulator and one vector accumulator are needed with the other partial product accumulators being reused. We illustrate the layout regularity by showing the floor planning of the macrocell design which accepts two input vectors of 16 elements with bit width of 16. Novel high performance low power building blocks such as full adder, 4-2 and 5-2 compressors presented in Chapters 3 and 4 are employed to further

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

enhance the performance of the scalar product core. Theoretical estimation and simulation results based on the model established on the physically extracted parameters from these leaf cells indicate that the proposed architecture is able to reduce the overall power, area and delay requirements to a greater extent than a conventional architecture. Since it is essentially a combinational design, the proposed macrocell is suitable for implementation as either a self-timed core or a synchronous multicycle core with an appropriate bus interface to high-speed processors. With the issue of only a single instruction, the macrocell can execute vector operations autonomously, effectively reducing both the instruction count and cycles per instruction factors of the processor. Very often, it is necessary to add some additional circuits to perform self-testing, error detection and correction, etc. These circuit overheads will cause the design to occupy more silicon area and incur longer delay. It is reported that some self-checking design for a VLSI processor adder occupied 13% of the area and 7% of the worst case delay [SHI91]. To facilitate the evaluation of the real performance of the core submitted for chip fabrication, the auxiliary on-chip circuits are also designed for testing the worst case delay under the restricted number of IO pins. A new built-in unique delay detection circuit with data-dependent delay error compensation is also proposed for this purpose.

This chapter is organized as follows: in Section 5.2, the new algorithm for scalar product computation is proposed. Based on the algorithm, the architecture, floor planning and delay estimation of scalar product IP core are detailed respectively in Sections 5.3, 5.4 and 5.5. The conventional architecture is compared with the proposed architecture in those sections. The layout of the proposed architecture and the design of the auxiliary circuit for the scalar product IP core in compliance with the CSM 0.18μm CMOS process technology are presented in Sections 5.6 and 5.7. This is followed by the comparison of the pre- and post- layout simulation results in Section 5.8. The chapter is concluded with a summary in Section 5.9.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

5.2 Algorithm of scalar product

Let $\mathbf{A} = [a_0, a_1, \ldots, a_{N-1}]$ and $\mathbf{X} = [x_0, x_1, \ldots, x_{N-1}]^T$ be two N-element vectors. The scalar product of $\mathbf{AX}$ is computed by

$$y = \mathbf{AX} = \begin{bmatrix} a_0 & a_1 & \cdots & a_{N-1} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{bmatrix} = \sum_{n=0}^{N-1} a_n x_n \tag{5.1}$$

On a standard processor that does not have a vector multiplication instruction, the scalar product computation is obtained by a series of programmed multiply-accumulate instructions issued sequentially. The computation of (5.1) can be sped up on hardware by streaming the flow of the operands for pipelined operation. In synchronous array implementation, clock distribution problem and clock-skews due to different clock path lengths and loads need to be addressed, whereas in a self-timed design, the queue capacity have to be optimized to maintain the desired pipelining period to avoid deadlock. We now develop an algorithm to accelerate the computation of (5.1) in a fully combinational structure in order to reduce the number of intermediate stages and solve the clock distribution problem.

In what follows, we will consider the basic algorithm for VLSI implementation of (5.1) in a traditional approach. The elements of both the vectors **A** and **X** are represented in 2's complement form. Their word lengths are assumed to be R and S bits, respectively.

$$a_n = -(a_n)_{R-1} 2^{R-1} + \sum_{r=0}^{R-2} (a_n)_r 2^r \tag{5.2}$$

and

$$x_n = -(x_n)_{S-1} 2^{S-1} + \sum_{s=0}^{S-2} (x_n)_s 2^s \tag{5.3}$$

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

where $(a_n)_r$ is the rth bit of a_n and $(x_n)_s$ is the sth bit of x_n. $(a_n)_{R-1}$ and $(x_n)_{S-1}$ are the most significant bits (MSBs) of a_n and x_n, respectively.

Substituting (5.2) and (5.3) into (5.1), we obtain (5.4) and (5.5)

$$\begin{aligned} y &= \sum_{n=0}^{N-1}\left[-(a_n)_{R-1}2^{R-1}+\sum_{r=0}^{R-2}(a_n)_r 2^r\right]\left[-(x_n)_{S-1}2^{S-1}+\sum_{s=0}^{S-2}(x_n)_s 2^s\right] \\ &= \sum_{n=0}^{N-1}\Bigg[(a_n)_{R-1}(x_n)_{S-1}2^{R+S-2}+\sum_{r=0}^{R-2}\sum_{s=0}^{S-2}(a_n)_r(x_n)_s 2^{r+s} \\ &\quad -(a_n)_{R-1}2^{R-1}\sum_{s=0}^{S-2}(x_n)_s 2^s-(x_n)_{S-1}2^{S-1}\sum_{r=0}^{R-2}(a_n)_r 2^r\Bigg] \end{aligned} \tag{5.4}$$

$$\begin{aligned} &= \sum_{n=0}^{N-1}\Bigg[(a_n)_{R-1}(x_n)_{S-1}2^{R+S-2}+\sum_{r=0}^{R-2}\sum_{s=0}^{S-2}(a_n)_r(x_n)_s 2^{r+s} \\ &\quad +\sum_{s=0}^{S-2}\left(1-(a_n)_{R-1}(x_n)_s\right)2^{s+R-1}+\sum_{r=0}^{R-2}\left(1-(a_n)_r(x_n)_{S-1}\right)2^{r+S-1} \\ &\quad -\sum_{r=S-1}^{R+S-3}2^r-\sum_{s=R-1}^{R+S-3}2^s\Bigg] \end{aligned} \tag{5.5}$$

Using the identity $\sum_{i=j}^{k-1}2^i = 2^k - 2^j$, the last two items of the constant accumulation are given by:

$$-\sum_{r=S-1}^{R+S-3}2^r-\sum_{s=R-1}^{R+S-3}2^s = -2^{R+S-1}+2^{S-1}+2^{R-1} \tag{5.6}$$

These constant bits in (5.6) are the correction bits used to reduce the gate counts that are otherwise required for the sign extension bits of each partial product terms. Collectively, they are termed the correction vector.

Thus, (5.5) can be expressed as follows:

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

$$y=\sum_{n=0}^{N-1}\Big[(a_n)_{R-1}(x_n)_{S-1}2^{R+S-2}+\sum_{r=0}^{R-2}\sum_{s=0}^{S-2}(a_n)_r(x_n)_s2^{r+s} +\sum_{s=0}^{S-2}\overline{(a_n)_{R-1}(x_n)_s}2^{s+R-1}+\sum_{r=0}^{R-2}\overline{(a_n)_r(x_n)_{S-1}}2^{r+S-1} -2^{R+S-1}+2^{S-1}+2^{R-1}\Big] \tag{5.7}$$

A scalar product IP core as implied by (5.7) requires a series of integer multipliers to produce the intermediate results and an accumulator to add them up. As an example, a 4×4 operands scalar product core with word length of four is shown in Fig. 5.1. Each symbol "*" denotes a partial product bit, and the "*"s in the shaded region denote the bit to be complemented. The symbols "p" enclosed in rectangular box denote the intermediate result bits generated by each of the individual multipliers operating in parallel. They are further summed to produce the final result bits as denoted by a rectangular box of symbols "r".

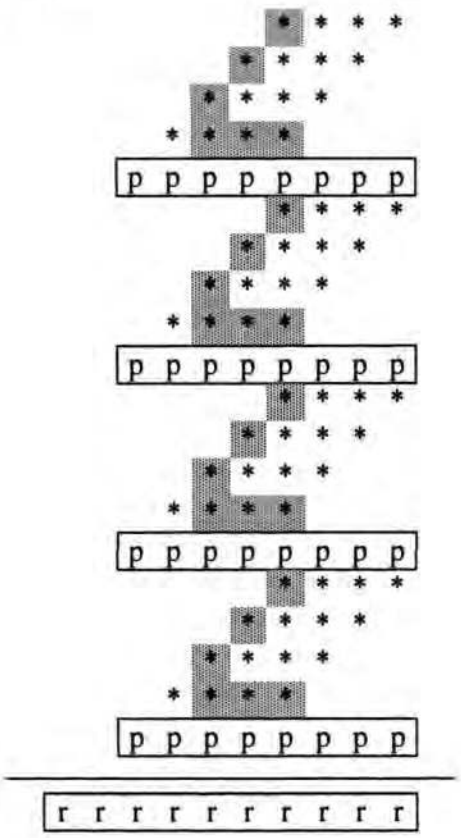


Figure 5.1 *Traditional algorithm for VLSI design of scalar product IP core*

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

The layout irregularity of the scalar product IP core obtained using the traditional approach often leads to an excessively low efficiency in the VLSI area usage. Also, the interconnecting wires are long, which cause longer delay in the critical path and a higher power dissipation.

We propose an alternative approach to the design of a highly modular vector multiplier. The method is based on the decomposition of the signed integer vector **A** in (5.1) into N weighted Boolean vectors. Weighted products of the input array elements gated by each Boolean vector elements are generated followed by a carry-free accumulation. The advantage of this decomposition algorithm is that the carry-propagation delay occurs only once at the very end, instead of at each addition step. This will enhance the opportunity to explore for a high density design with a more granular layout.

From (5.4):

$$
\begin{aligned}
y &= \sum_{n=0}^{N-1}\Bigg[-(x_n)_{S-1}2^{S-1}\sum_{r=0}^{R-2}(a_n)_r 2^r + \sum_{r=0}^{R-2}\sum_{s=0}^{S-2}(a_n)_r(x_n)_s 2^{r+s} \\
&\quad + (a_n)_{R-1}(x_n)_{S-1}2^{R+S-2} - (a_n)_{R-1}2^{R-1}\sum_{s=0}^{S-2}(x_n)_s 2^s\Bigg] \\
&= \sum_{n=0}^{N-1}\Bigg\{\sum_{r=0}^{R-2}\left(1-(a_n)_r(x_n)_{S-1}\right)2^{r+S-1} + \sum_{r=0}^{R-2}\sum_{s=0}^{S-2}(a_n)_r(x_n)_s 2^{r+s} \\
&\quad - \Bigg[\left(1-(a_n)_{R-1}(x_n)_{S-1}\right)2^{R+S-2} + (a_n)_{R-1}2^{R-1}\sum_{s=0}^{S-2}(x_n)_s 2^s - 2^{R+S-2}\Bigg] \\
&\quad - \sum_{r=S-1}^{R+S-3}2^r\Bigg\} \\
&= \sum_{n=0}^{N-1}\Bigg\{\sum_{r=0}^{R-2}\overline{(a_n)_r(x_n)_{S-1}}\,2^{r+S-1} + \sum_{r=0}^{R-2}\sum_{s=0}^{S-2}(a_n)_r(x_n)_s 2^{r+s} \\
&\quad - \Bigg[\overline{(a_n)_{R-1}(x_n)_{S-1}}\,2^{R+S-2} + (a_n)_{R-1}2^{R-1}\sum_{s=0}^{S-2}(x_n)_s 2^s\Bigg] \\
&\quad + 2^{R+S-2} - \sum_{r=S-1}^{R+S-3}2^r\Bigg\}
\end{aligned}
\tag{5.8}
$$

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

A regular array of accumulation units will not only utilize the silicon area more efficiently, but also reduce the glitches caused by the unequal delay paths of the input signals to the accumulation units. In order to obtain a regular layout of the accumulation units, we swap the accumulating order of r and n.

$$\begin{aligned} y &= \sum_{r=0}^{R-2}\left\{\sum_{n=0}^{N-1}\left[\overline{(a_n)_r(x_n)_{S-1}}2^{r+S-1}+\sum_{s=0}^{S-2}(a_n)_r(x_n)_s2^{r+s}\right]\right\} \\ &\quad -\sum_{n=0}^{N-1}\left[\overline{(a_n)_{R-1}(x_n)_{S-1}}2^{R+S-2}+2^{R-1}\sum_{s=0}^{S-2}(a_n)_{R-1}(x_n)_s2^s\right]+\sum_{n=0}^{N-1}2^{S-1} \\ &= \sum_{r=0}^{R-2}\left\{2^r\sum_{n=0}^{N-1}\left[\overline{(a_n)_r(x_n)_{S-1}}2^{S-1}+\sum_{s=0}^{S-2}(a_n)_r(x_n)_s2^s\right]\right\} \\ &\quad -2^{R-1}\sum_{n=0}^{N-1}\left[\overline{(a_n)_{R-1}(x_n)_{S-1}}2^{S-1}+\sum_{s=0}^{S-2}(a_n)_{R-1}(x_n)_s2^s\right]+\sum_{n=0}^{N-1}2^{S-1} \end{aligned} \tag{5.9}$$

The width of x_n is S bits. Let $\xi = \lceil \log_2 N \rceil$ be the smallest integer larger than or equal to $\log_2 N$. The expressions in both the positive and negative terms of (5.9):

$$\sum_{n=0}^{N-1}\left[\overline{(a_n)_r(x_n)_{S-1}}2^{S-1}+\sum_{s=0}^{S-2}(a_n)_r(x_n)_s2^s\right] \qquad r=0,1,\cdots,R-1 \tag{5.10}$$

are the unsigned numbers, each of which can always be represented in exactly $(S+\xi)$ bits in normal unsigned binary format, where the rth expression for $r = 0, 1, \ldots R-1$ is given by $\sum_{i=0}^{S+\xi-1}(z_r)_i \cdot 2^i$. However, in order to avoid the long delay to generate the intermediate results, we use the stored carry format instead of the normal unsigned binary format for their representation. The stored carry format of the intermediate result always has the formation shown in Fig. 5.2, where one row is the sum and the other row is the stored carry. The MSBs of the sum and the stored carry are both located at column numbers $2^{S+\xi-2}$. The binary value represented by the stored sum and carry is equal to the unsigned binary number representation given in the third row.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

$$\begin{array}{llllllll} & s & s & s & \cdots\cdots\cdots & s & s & s \\ + & c & c & c & \cdots\cdots\cdots & c & c & \\ \hline z & z & z & z & \cdots\cdots\cdots & z & z & z \end{array}$$

Figure 5.2 *Stored carry format*

Thus, (5.10) can be expressed in stored carry format as follows.

$$\begin{aligned} &\sum_{n=0}^{N-1}\left[\overline{(a_n)_r(x_n)_{S-1}}2^{S-1}+\sum_{s=0}^{S-2}(a_n)_r(x_n)_s 2^s\right] \\ &=\sum_{i=0}^{S+\xi-2}(s_r)_i 2^i+\sum_{i=1}^{S+\xi-2}(c_r)_i 2^i \qquad r=0,1,\cdots,R-1 \end{aligned} \tag{5.11}$$

where $(s_r)_i$ and $(c_r)_i$ are the ith bits of stored sum and stored carry used to represent the rth expression of (5.10).

Substituting (5.11) into (5.9), we have

$$\begin{aligned} y=&\sum_{r=0}^{R-2}2^r\left(\sum_{i=0}^{S+\xi-2}(s_r)_i 2^i+\sum_{i=1}^{S+\xi-2}(c_r)_i 2^i\right) \\ &-2^{R-1}\left(\sum_{i=0}^{S+\xi-2}(s_{R-1})_i 2^i+\sum_{i=1}^{S+\xi-2}(c_{R-1})_i 2^i\right)+\sum_{n=0}^{N-1}2^{S-1} \\ =&\sum_{r=0}^{R-2}2^r\left(\sum_{i=0}^{S+\xi-2}(s_r)_i 2^i+\sum_{i=1}^{S+\xi-2}(c_r)_i 2^i\right) \\ &+2^{R-1}\left[\sum_{i=0}^{S+\xi-2}\left(1-(s_{R-1})_i\right)2^i+\sum_{i=1}^{S+\xi-2}\left(1-(c_{R-1})_i\right)2^i\right] \\ &+N2^{S-1}-2^{R-1}\left(2^{S+\xi-1}-1+2^{S+\xi-1}-2\right) \\ =&\sum_{r=0}^{R-2}2^r\left(\sum_{i=0}^{S+\xi-2}(s_r)_i 2^i+\sum_{i=1}^{S+\xi-2}(c_r)_i 2^i\right) \\ &+2^{R-1}\left[\sum_{i=0}^{S+\xi-2}\overline{(s_{R-1})_i}2^i+\sum_{i=1}^{S+\xi-2}\overline{(c_{R-1})_i}2^i\right] \\ &+N2^{S-1}-2^{R+S+\xi-1}+2^{R-1}+2^R \end{aligned} \tag{5.12}$$

The correction vector in (5.12) is $N\cdot 2^{S-1}-2^{R+S+\xi-1}+2^{R-1}+2^R$.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Comparing (5.7) and (5.12), we found that (5.7) needs $N(S-1) + N(R-1)$ inverters to complement the bits while (5.12) needs $(S+\xi-1) + (S+\xi-2) + NR$ inverters. To illustrate the significance of its implication, consider the case of $N = 16$ and $S = R = 16$, then $\xi = \log_2 N = 4$, (5.7) requires 480 inverters whereas (5.12) requires only 293 inverters, which represents a saving of 39% of inverters and potential spurious transitions.

In terms of VLSI layout, every arithmetic module is regular in shape. Using the same example and notations as in Fig. 5.1, the structural differences of the proposed method are shown in Fig. 5.3.

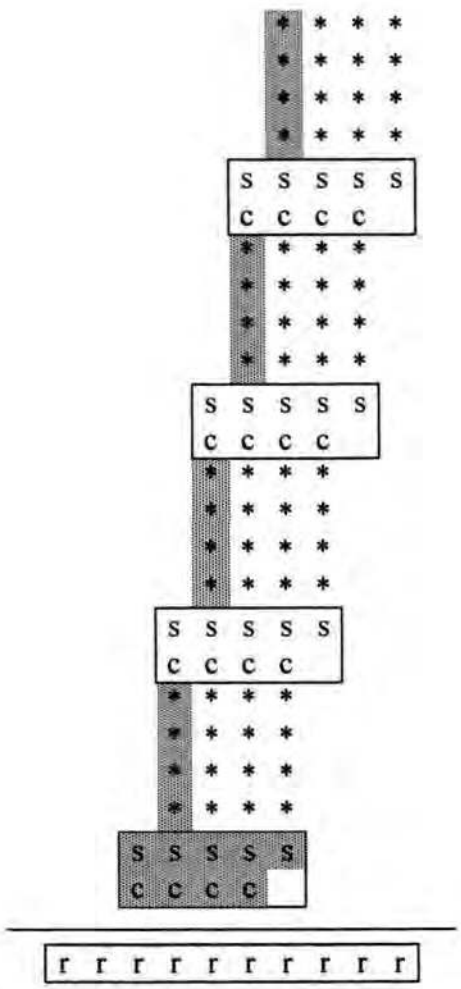


Figure 5.3 *Proposed algorithm for VLSI design of vector multiplier*

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

5.3 Architectural design of the scalar product core

Previous attempts to design a dedicated macrocell for accelerating the computation of scalar product have often been thwarted by the limitation of silicon area and restricted routing channels available for the interconnecting wires. This and the following sections address the unique advantages of our algorithm when it is mapped into hardware architecture, particularly when the design is to be implemented in deep-submicron technology where the signal propagations and transitions are dominated by the interconnect delay. We will consider the design example of a scalar product macrocell, operating on two input vectors. Each input vector consists of 16 signed integer elements in two's complement form. The word length of each element is 16. In other words, N = 16, and $R = S = 16$ for (5.12).

The architecture of the scalar product core is mainly composed of three parts, namely the partial product generator, the partial product accumulator, and the vector accumulator, as shown in Fig. 5.4. For convenience, they are abbreviated as PPG, PPA and VA, respectively.

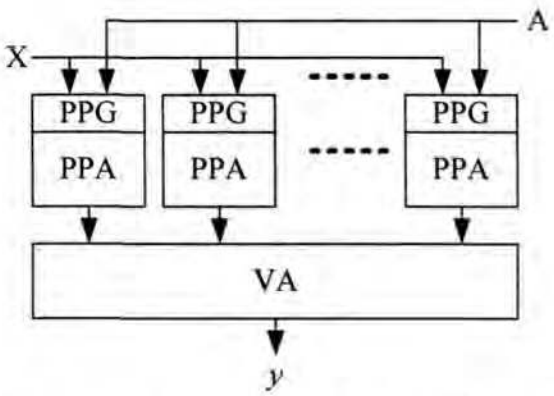


Figure 5.4 *Proposed architecture for the scalar product core*

The PPG generates the partial products from the input vector **X** according to the bit value of $(a_n)_r$. It is composed of a row of AND gates, except for the most significant bit, which is generated by a NAND gate because the MSB needs to be complemented.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

The PPA sums up all the partial products of the same weight and generates the intermediate result. This makes the accumulating circuit form a rectangle structure, as shown in Fig. 5.5. Such structure permits optimization of routing channels with short equal interconnects between neighboring arithmetic cells.

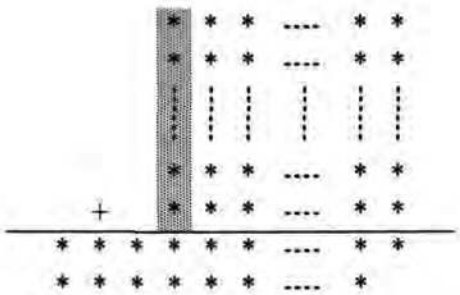


Figure 5.5 *Rectangular structure of PPA*

The accumulator is realized by a Wallace tree structure [PAR00, VAI01, WAL64, DAD65], which consists of three layers of (4,2) compressors [PAR00, VAI01]. Instead of summing to a final result that often needs a time-consuming carry propagation adder (CPA), every accumulator just produces a stored carry intermediate result generated by the third layer of (4,2) compressors, as shown in Fig. 5.6. The elimination of the internal CPAs by keeping the intermediate sum in stored carry format will reduce the computation time and glitches.

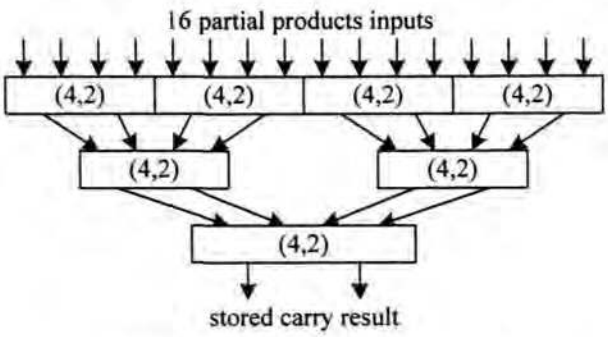


Figure 5.6 *Wallace tree structure of PPA*

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

The VA sums up all the 16 stored carry intermediate results generated by the PPA's, which is equivalent to 32 binary numbers. These stored carry numbers are weighted differently and their dot notation forms a parallelogram as shown in Fig. 5.7. All the bits in the stored carry sum of highest weight are negated before the accumulation. The letter "e" in the figure denotes the correction bits, which is $2^{35}+2^{19}+2^{16}+2^{15}$. Compared with the other 16 regular PPAs, this single irregularity of VA is trivial. Moreover, the 16 PPAs cover most area of the chip, and the arithmetic circuits of the VA are interleaved uniformly over the PPAs (refer to the floor planning in Fig. 5.10 of Section 5.4), making the irregularity almost indiscernible.

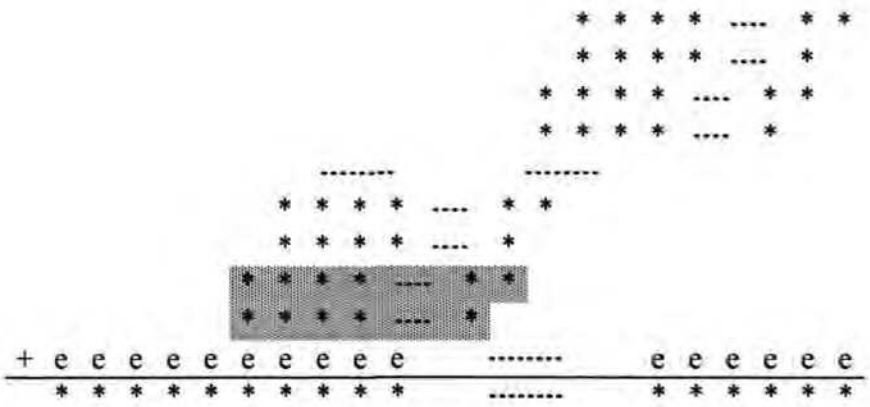


Figure 5.7 *Parallelogram structure of VA*

The vector accumulator also employs the Wallace tree structure with four layers of (4,2) compressors, as shown in Fig. 5.8. The correction value is added to the last layer of (4,2) compressors to form the (5,2) compressors. The final 36-bit result is computed by a CPA.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

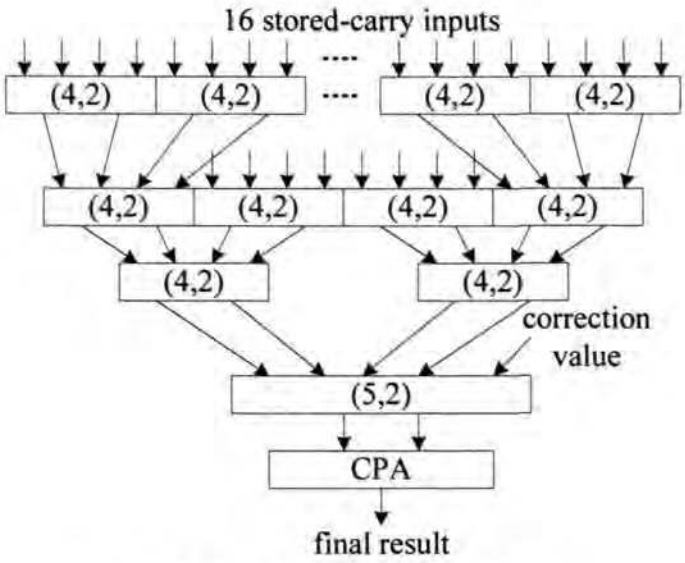


Figure 5.8 *Wallace tree structure of VA*

5.4 Floor planning of the scalar product core

Owing to the large number of transistors involved in the design, it is extremely difficult to make a custom layout of the entire circuit. Fortunately, the architecture of the scalar product macrocell is highly modular, making it possible to handcraft the physical design hierarchically. We use the leaf cells presented in the previous chapters to assemble them into functional blocks. The functional blocks are abutted to form the complete circuit. The goal of our floor planning is to make the interconnecting wires as short as possible and the difference in the lengths of the interconnecting wires of adjacent layers of the Wallace tree as small as possible.

The main leaf cells of the core are (4,2) compressors, (5,2) compressors, full adders, 4-bit carry look ahead adders, and NAND and AND gates for the PPG. In order to make a quality floor planning, we list the sizes of these leaf cells prototyped on CSM 0.18μm process technology in Table 5.1, most of which have already been introduced in Chapters 3 and 4.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Table 5.1 *The sizes of different leaf cells (on CSM 0.18μm CMOS process)*

Leaf Cell	**Width,** w (μm)	**Height,** h (μm)	**Area,** a (μm^2)
(4,2) compressor	24.2	17	411.4
(5,2) compressor	31.4	17	533.8
full adder	12	10	120
CLA4	21.3	32.3	687.99
AND	4.6	4.2	19.32

The floor planning of the PPA with the PPG is shown in Fig. 5.9. Every rectangular block labeled with a number is a column of (4,2) compressors. The number refers to the layer number of the block in the Wallace tree. For example, the first layer of blocks labeled "1" comprises those blocks whose inputs are fed directly from the PPG. The second layer of blocks labeled "2" receives its inputs from the outputs of the nearest left and right blocks labeled "1". Thus, Block "1" has 16 (4,2) compressors, Block "2" 17 compressors, and Block "3" 18 compressors. Their areas are given by $S_{PPA1} = w_{42} \times 16h_{42}$, $S_{PPA2} = w_{42} \times 17h_{42}$ and $S_{PPA3} = w_{42} \times 18h_{42}$, respectively.

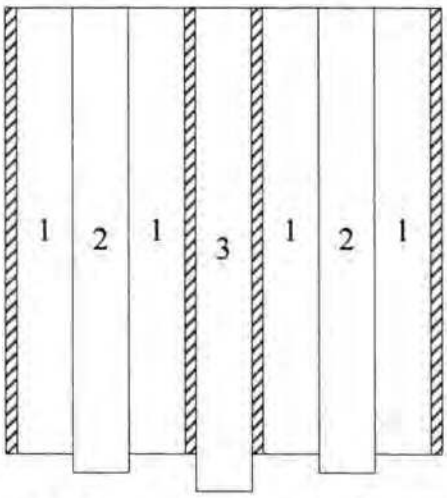


Figure 5.9 *Floor planning of PPA with PPG*

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

The PPG (the thin strips marked with cross hatch lines on Fig. 5.9) are arranged along one side of Block "1" of the PPA. This is because the four input pads of (4,2) compressor are located vertically along one side of Block "1" and the height of every four AND gates are almost equal to that of a (4,2) compressor. In this way the length of the connecting wires are minimized from PPG to PPA Block "1". Every Block "1" needs four 16-bit partial product generators, which is equivalent to 64 elements, the size of which is $S_{PPG64} = w_{AND} \times 64h_{AND} \approx w_{AND} \times 16h_{42}$.

As mentioned earlier, Block "2" is sandwiched between two adjacent banks of Block "1", and its outputs are connected to Block "3". Similarly, Block "3" is interpolated such that it is equi-distance from two nearest left and right banks of Blocks "2" in view of the large number of wires connecting Blocks "1" and "2", and Blocks "2" and "3". Such an arrangement would reduce the wire lengths as much as possible, minimizing the switching power and glitches caused by the stray capacitance of the wires. The area efficient rectangular formation of the PPA is a result of mapping the proposed algorithm.

The PPA with PPG in Fig. 5.9 covers a rectangular area of $S_{PPA} = w_{PPA} \times h_{PPA} = (7w_{42} + 4 w_{AND}) \times 18h_{42} = 187.8 \times 306 = 57466.8\ \mu m^2$. However, this rectangular area is not fully utilized. Only the area of $S_{PPA}' = 4S_{PPG64} + 4S_{PPA1} + 2S_{PPA2} + S_{PPA3} = 52727.2\ \mu m^2$ is covered by the leaf cells. The VLSI area usage ratio of the PPA is therefore:

$$\rho_{PPA} = \frac{S_{PPA}}{S_{PPA}'} = \frac{52727.2}{57466.8} = 91.8\% \qquad (5.13)$$

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

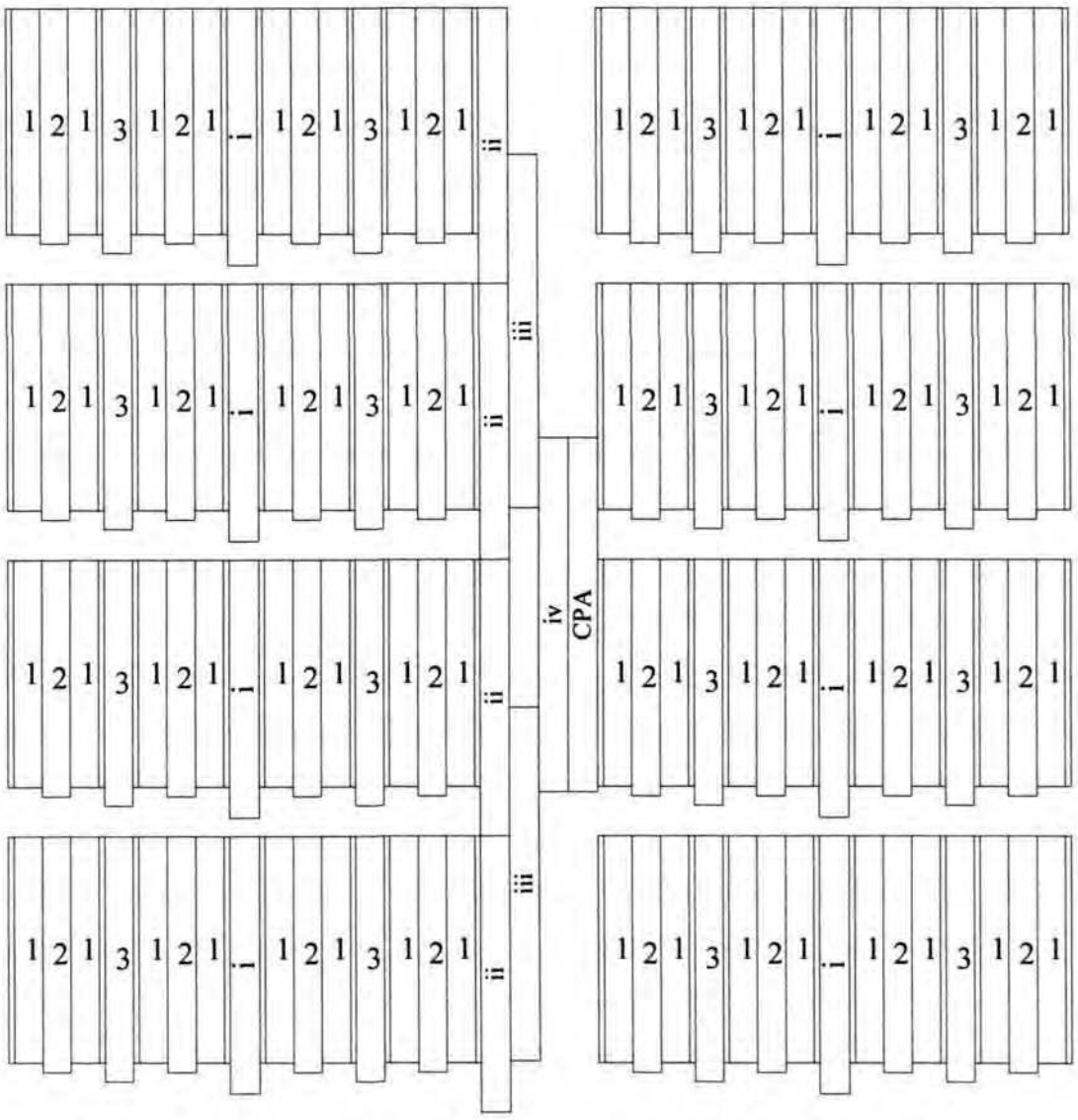


Figure 5.10 *Floor planning of the proposed bit-parallel scalar product core*

The scalar product core needs 16 such PPAs with PPGs together with one VA. The complete floor planning of the macrocell is shown in Fig. 5.10. The roman number labelled blocks belong to the VA, where the arabic numbers refer to the layer numbers of the PPA blocks in the Wallace tree.

The width of Layer "i" is 19 bits, Layer "ii" 21 bits and Layer "iii" 24 bits. Layer "iv" is of 32 bits, which is implemented by the (5,2) compressors. The CPA layer is of 32 bits.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Therefore, the areas occupied by the abovementioned layers are $S_{VAi} = w_{42} \times 19h_{42}$, $S_{VAii} = w_{42} \times 21h_{42}$, $S_{VAiii} = w_{42} \times 24h_{42}$, $S_{VAiv} = w_{52} \times 19h_{52}$, and $S_{CPA} = w_{CLA4} \times 8h_{CLA4}$.

The proposed scalar product core needs a total rectangular silicon area of:

$$S_{SPC\text{-}1} = w_{SPC\text{-}1} \times h_{SPC\text{-}1} = (4w_{PPA} + 4w_{42} + w_{52} + w_{CLA4}) \times (4 \times 21h_{42}) = 900.7 \times 1428 = 1286199.6\ \mu m^2. \quad (5.14)$$

Of which the physically used area is

$$S_{SPC\text{-}1}' = 16S_{PPA}' + 8S_{VAi} + 4S_{VAii} + 2S_{VAiii} + S_{VAiv} + S_{CPA} = 976118.92\ \mu m^2. \quad (5.15)$$

Therefore, the overall area usage efficiency of the scalar product core is:

$$\rho_{SPC\text{-}1} = \frac{S_{SPC\text{-}1}'}{S_{SPC\text{-}1}} = \frac{976118.92}{1286199.6} = 75.9\%. \quad (5.16)$$

To make a fair comparison and to illustrate the advantage of our proposed method, the architecture offered by the conventional method is laid out using the same leaf cells with the same floor planning goal. A 16×16-bit standard multiplier is the main building block of the straightforward approach to implement the architecture of the scalar product macrocell. Its layout is shown in Fig. 5.11. The shaded blocks are PPG.

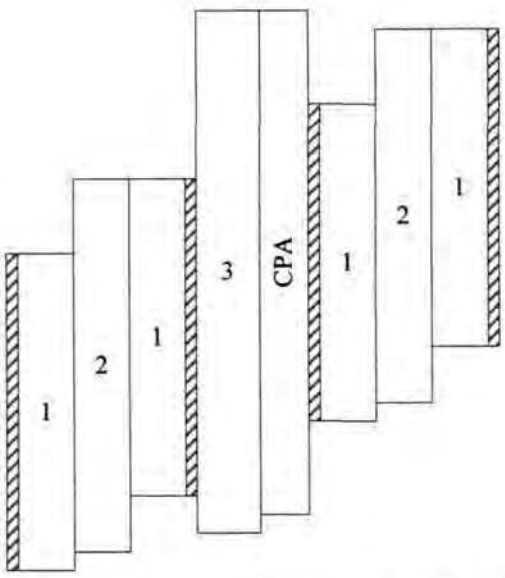


Figure 5.11 *Floor-planning of normal 16×16-bit multiplier*

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Layer "1" is of 17 bits length, composed of 3 full adders and 14 (4,2) compressors. Layer "2" has 21 bits length, composed of 8 full adders and 13 (4,2) compressors. Layer "3" 29 bits, composed of 16 full adders and 13 (4,2) compressors. Layer "CPA" has 28 bits. Their areas are denoted as S_1, S_2, S_3 and $S_{CPA\text{-}1}$, respectively. It is evident that the layout of the multiplier is irregular if the aim of the design is to minimize the delay of the interconnecting wires. Regulating the layout will make both the local and global interconnecting wires longer and their routings interlace irregularly, thus increasing the imparity of the signal arrival time and glitch power.

The multiplier in Fig. 5.11 covers a rectangular area of:

$$S_{MUL16} = w_{MUL16} \times h_{MUL16} = (7w_{42} + 4w_{AND} + w_{CLA4}) \times 30h_{42} = 209.1 \times 510 = 106641 \ \mu m^2. \quad (5.17)$$

The physically used area is

$$S_{MUL16}' = 4S_{PPG} + 4S_1 + 2S_2 + S_3 + S_{CPA\text{-}1} = 4 \times 16 \times 4S_{AND} + 4(3S_{FA} + 14S_{42}) + 2(8S_{FA} + 13S_{42}) + (16S_{FA} + 13S_{42}) + 7S_{CLA4} = 54124.85 \ \mu m^2. \quad (5.18)$$

Therefore, the area usage efficiency of the multiplier is:

$$\rho_{MUL16} = \frac{S_{MUL16}'}{S_{MUL16}} = \frac{54124.85}{106641} = 50.8\%. \quad (5.19)$$

The floor planning of the entire macrocell of the conventional architecture is shown in Fig. 5.12. The roman number labeled blocks perform the accumulation of all the intermediate products. Block "i" is of 32 bits, Block "ii" 33 bits and Block "iii" 34 bits, all of which are implemented using (4,2) compressors. The CPA block besides Block "iii" is of 36 bits. Their areas are denoted as S_i, S_{ii}, S_{iii} and $S_{CPA\text{-}2}$, respectively.

The whole conventional scalar product macrocell covers a rectangular area of:

$$S_{SPC\text{-}2} = w_{SPC\text{-}2} \times h_{SPC\text{-}2} = (4w_{MUL16} + 3w_{42} + w_{CPA\text{-}2}) \times (4 \times 32h_{42}) = 930.3 \times 2176 = 2024332.8 \ \mu m^2. \quad (5.20)$$

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

The physically used area is:

$$S_{SPC\text{-}2}' = 16S_{MUL16}' + 4S_i + 2S_{ii} + S_{iii} + S_{CPA\text{-}2} = 16S_{MUL16}' + 4 \times 32S_{42} + 2 \times 33S_{42} + 34S_{42} + 9S_{CLA4} = 965988.71\ \mu m^2. \quad (5.21)$$

Therefore the VLSI area usage efficiency of the conventional scalar product macrocell is:

$$\rho_{SPC\text{-}2} = \frac{S_{SPC\text{-}2}'}{S_{SPC\text{-}2}} = \frac{965988.71}{2024332.8} = 47.7\%. \quad (5.22)$$

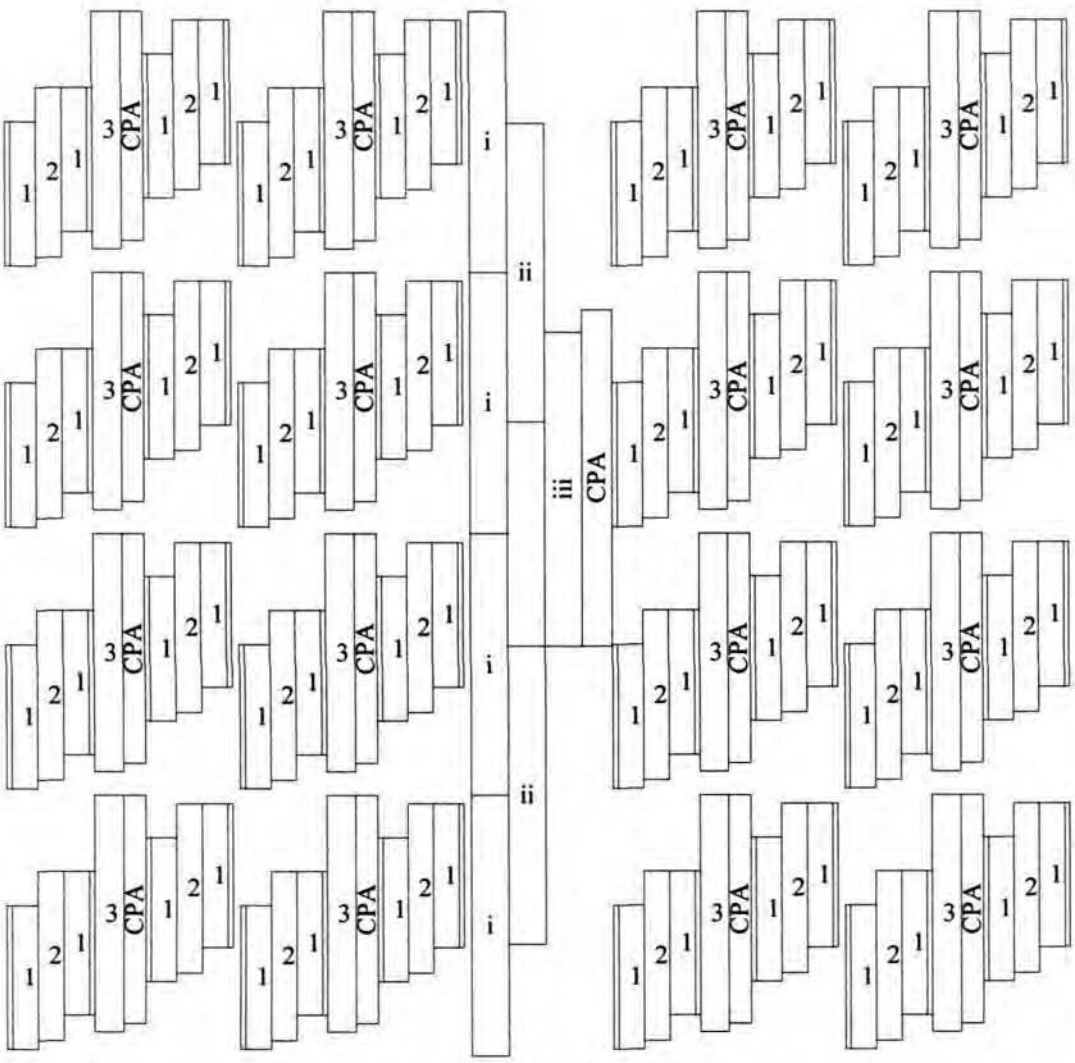


Figure 5.12 *Floor planning of the conventional scalar product macrocell*

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

The layout characteristics of the conventional and proposed architectures are summarized in Table 5.2. It is evident that the proposed architecture has advantages over the conventional one. There is a saving of 36.5% of silicon area and up to 59% increase in terms of the efficiency of the area usage.

Table 5.2 *Floor planning characteristics of the two architectures*

	Proposed	**Conventional**
Layout Area Width (μm)	900.7	930.3
Layout Area Height (μm)	1428	2176
Total Area (μm^2)	1286199.6	2024332.8
Area Efficiency	75.9 %	47.7 %
Regularity of Blocks	Regular	Irregular

5.5 Delay estimation based on floor planning

The computation time or the delay is mainly caused by the gate delay and the interconnection delay [KAT00, SYL98, NAN99]. For both architectures, there is not much difference in the gate delay. However, the interconnection delay, especially the one in the vector accumulator, varies significantly. In what follows, we model theoretically the critical delay from the inputs of the PPG to the generation of the final result.

The gate delay of the proposed architecture is composed of three parts, namely the delays for the PPG, PPA and VA. The gate delay of the PPG is given by $t_{PPG} = t_{AND}$. Since PPA has $[(\log_2 n) - 1]$ layers of (4,2) compressors, its gate delay is $t_{PPA} = [(\log_2 n) - 1]t_{42}$, where n is the word length of the element of the input vector, t_{42} is the longest delay of the (4,2) compressor. Therefore,

$$t_{PPA} = 3t_{42}. \tag{5.23}$$

The critical path of VA is composed of three (4,2) compressors in the first three layers, one (5,2) compressor in the fourth adder and two 32-bit length CLAs in the CPA layer. Therefore, the delay of the vector accumulator is given by:

$$t_{VA} = 3t_{42} + t_{52} + t_{CLA32}. \tag{5.24}$$

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

The interconnection delay is technology dependent. It is proportional to the stray capacitance of the connecting wires. Since the width of the wire δ is normally fixed by the feature size of the process, the capacitance is proportional to the length Λ of the wire. Therefore the interconnection delay is also proportional to Λ. For a first order estimation, let k be the constant of proportionality for a given technology, the delay of wire is given by:

$$t_{wire} = k\delta\Lambda. \tag{5.25}$$

The maximum length of the connecting wire in the critical path can be expressed as:

$$\Lambda_{maxim\text{-}1}=\Lambda_{PPG\text{-}1} + \Lambda_{1\text{-}2} + \Lambda_{2\text{-}3} + \Lambda_{3\text{-}i} + \Lambda_{i\text{-}ii} + \Lambda_{ii\text{-}iii} + \Lambda_{iii\text{-}iv} + \Lambda_{iv\text{-}CPA} \tag{5.26}$$

where the arabic and roman numbers refer to the layer numbers in the Wallace tree of the PPA and VA, respectively. For convenience, we assume that the interconnecting lines are linked between the centers of two connected blocks. From Fig. 5.10, $\Lambda_{PPG\text{-}1} = 0.5(w_{AND} + w_{42})$, $\Lambda_{1\text{-}2} = w_{42}$, $\Lambda_{2\text{-}3} = 2w_{42} + w_{AND}$, $\Lambda_{3\text{-}i} = 4w_{42} + 2w_{AND}$, $\Lambda_{i\text{-}ii} = w_{42} + w_{PPA}$, $\Lambda_{ii\text{-}iii} = w_{42} + 0.5h_{VAii}$, $\Lambda_{iii\text{-}iv} = 0.5(w_{42} + w_{52}) + h_{VAii}$ and $\Lambda_{iv\text{-}CPA} = 0.5(w_{52} + w_{CLA4})$.

Therefore $\Lambda_{maxim\text{-}1}$= 14.4 + 24.2 + 53 + 106 + 212 + 202.7 + 384.8 + 52.7 = 1049.8 μm.

The worst case delay of the proposed architecture is given by:

$$T_{SPC\text{-}1} = t_{gate\text{-}1} + t_{wire\text{-}1} = t_{PPG} + t_{PPA} + t_{VA} + k\delta\Lambda_{maxim\text{-}1}$$
$$= t_{AND} + 6t_{42} + t_{52} + t_{CLA32} + k\delta \times 1049.8\mu. \tag{5.27}$$

In the conventional architecture, the critical path of a 16-bit multiplier is composed of one partial product generator, one (4,2) compressor each in the first, second and third layers, and one carry look ahead adder in the CPA layer of the multiplier.

The delay of a multiplier is given by:

$$t_{mul} = t_{AND} + 3t_{42} + t_{CLA4}. \tag{5.28}$$

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

The critical path in the intermediate product accumulator includes three layers of (4,2) compressors and 28 bit-length carry look ahead adders in the CPA layer. Its delay is

$$t_{PA} = 3t_{42} + t_{CLA28} \tag{5.29}$$

From Fig. 5.12, the interconnection delay can be estimated as follows:

$$\Lambda_{maxim\text{-}2} = \Lambda_{PPG\text{-}1} + \Lambda_{1\text{-}2} + \Lambda_{2\text{-}3} + \Lambda_{3\text{-}CPA1} + \Lambda_{CPA1\text{-}i} + \Lambda_{i\text{-}ii} + \Lambda_{ii\text{-}iii} + \Lambda_{iii\text{-}CPA2} \tag{5.30}$$

where $\Lambda_{PPG\text{-}1} = 0.5(w_{AND} + w_{42})$, $\Lambda_{1\text{-}2} = w_{42}$, $\Lambda_{2\text{-}3} = 2w_{42} + w_{AND}$, $\Lambda_{3\text{-}CPA1} = 0.5(w_{42} + w_{CLA4})$, $\Lambda_{CPA1\text{-}i} = w_{MUL16} + 3w_{42} + 2w_{AND} + 0.5w_{CLA4}$, $\Lambda_{i\text{-}ii} = w_{42} + 0.5h_i$, $\Lambda_{ii\text{-}iii} = w_{42} + h_i$ and $\Lambda_{iii\text{-}CPA2} = 0.5(w_{42} + w_{CLA4})$.

Thus, we have

$$\Lambda_{maxim\text{-}2} = 14.4 + 24.2 + 53 + 52.7 + 301.55 + 296.2 + 568.2 + 52.7 = 1362.95\ \mu m.$$

Therefore, the worst case delay of the conventional architecture is given by:

$$T_{SPC\text{-}2} = t_{mul} + t_{PA} + t_{wire\text{-}2} = t_{AND} + 6t_{42} + t_{CLA4} + t_{CLA28} + k\delta \times 1362.95\mu. \tag{5.31}$$

The estimated worst case delay is shown in Table 5.3. The interconnect delay ratio of the proposed architecture to the conventional architecture is given by $\frac{k\delta \times 1049.8\mu}{k\delta \times 1362.95\mu} = 77\%$.

A saving 23% of the delay from the interconnecting wires is achieved. Although the gate delay is almost the same for the two architectures, the interconnect delay varies significantly. It should be noted that the good floor planning alternative and architecture in this case is a desirable outcome of the novel algorithm introduced in Section 5.2.

Table 5.3 *Estimation of the worst case delay*

	Proposed	**Conventional**
Gate Delay	$t_{AND} + 6t_{42} + t_{52} + t_{CLA32}$	$t_{AND} + 6t_{42} + t_{CLA4} + t_{CLA28}$
Interconnect Delay	$k\delta \times 1049.8\mu$	$k\delta \times 1362.95\mu$

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

5.6 The scalar product core layout in CSM 0.18μm technology

Based on the optimally floor planning of the proposed scalar product core, the entire chip is implemented on Chartered CSM 0.18μm 6-Metal 1-Poly CMOS 1.8V technology[1]. Fig. 5.13 shows the layout of one PPA, which validates the succinctness of the floor planning of the proposed architecture. The layout size of the PPA is 295μm in width by 308 μm in height while the earlier estimated size from the floor planning is 187.8μm by 306μm. The height is almost the same, but the error in the width estimation is notable. The discrepancy is due to the insertion of the data bus lines, and the local power and ground lines vertically between Blocks "1" and "3" instead of overlapping them with the PPA blocks. This is to reduce the amount of stray capacitance and minimize the interference with the data bus lines.

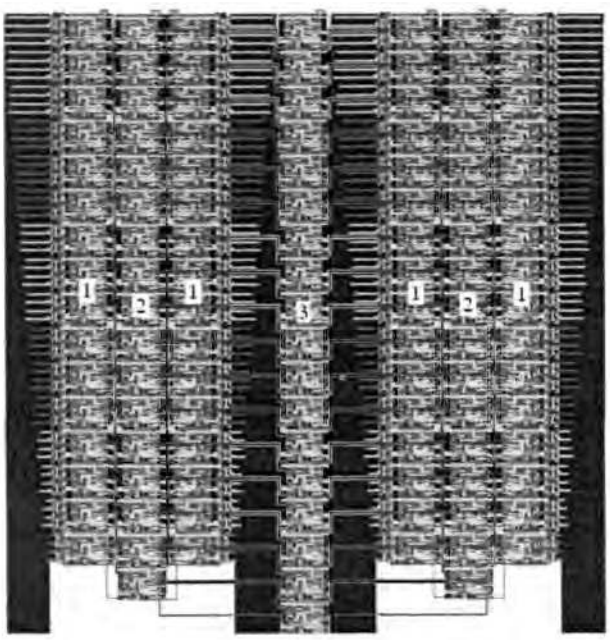


Figure 5.13 *Layout of the partial product accumulator*

Fig. 5.14 shows the complete chip layout of the proposed scalar product computation unit. It consists of the core for the scalar product calculation and some auxiliary circuits, such

[1] Chartered CSM 0.18μm CMOS process is closely compatible to the TSMC 0.18μm CMOS process, both in design rules and parasitic extraction. The use of Chartered CSM 0.18μm CMOS process is mainly due to the foundry accessibility for geographic reason at the time of preparing the layout.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

as the on-chip registers to hold the input operands and the output results, testing circuit for delay measurement, and IO pads. The blocks from VA-i to VA-iv and the peripheral circuits are labeled in the figure. The physical layout resembles the floor planning sketch of Fig. 5.10. The core circuit, containing 226560 transistors, occupies a rectangular silicon area of approximately 1430μm × 1560μm. The whole chip covers an area of 2600μm × 2900μm.

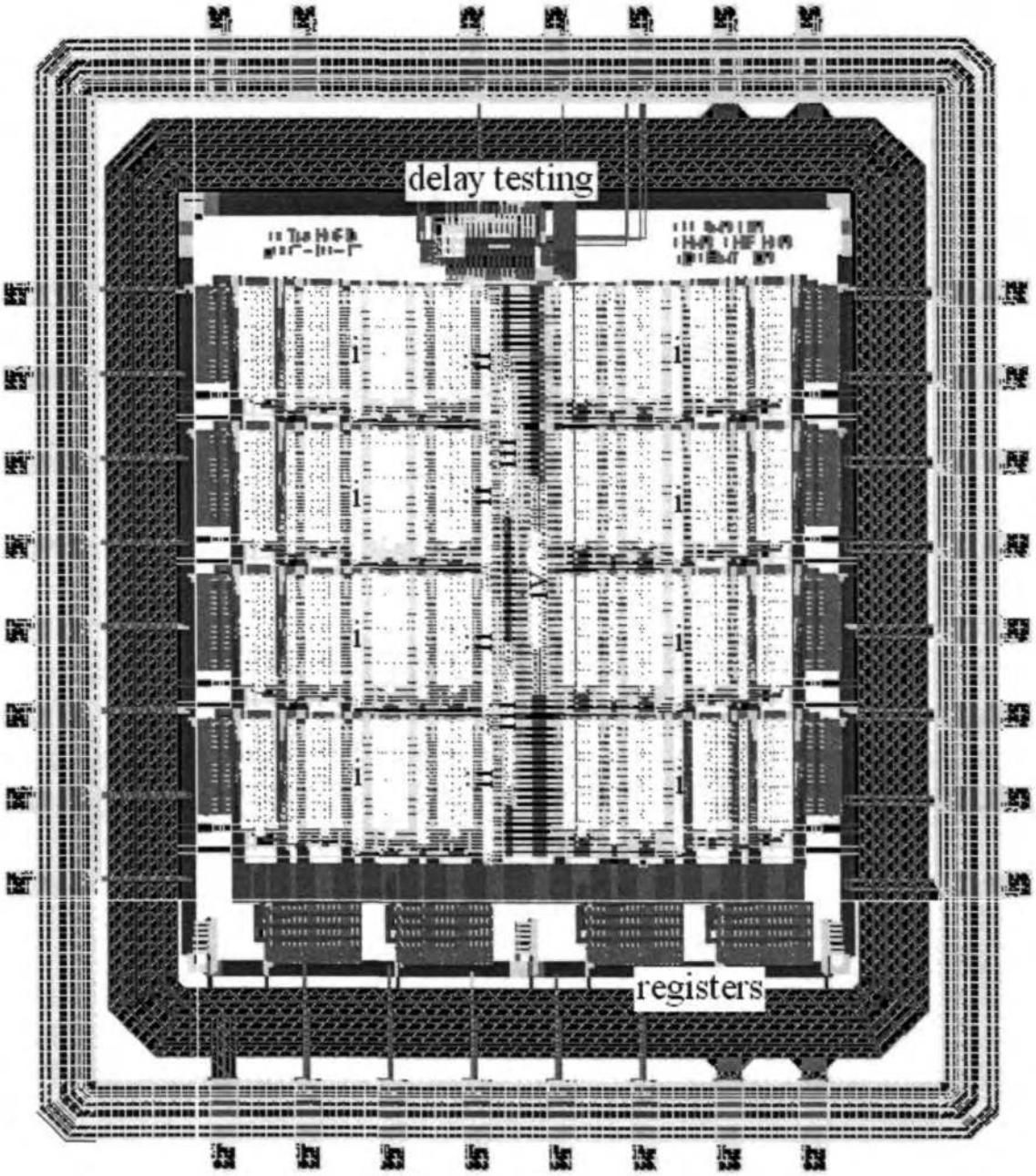


Figure 5.14 *Layout of the scalar product macrocell*

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

The six layers of metal and one layer of poly silicon are carefully planned for the interconnections. Metal-1, Metal-2 and Poly are used by the basic cells such as the compressors. For better conductivity, the only layer of Poly is never used for the inter-cell or global connecting lines. Metal-3 and Metal-4 are used for the interconnecting lines of the PPAs. These two layers of metals are also used for the interconnecting lines of the VAs, where they do not cross over the PPAs. Metal-5 is used to connect the PPA and the VA because these lines cross over the area where Metal-3 and Metal-4 are used. Almost all of the interconnecting lines are run horizontally.

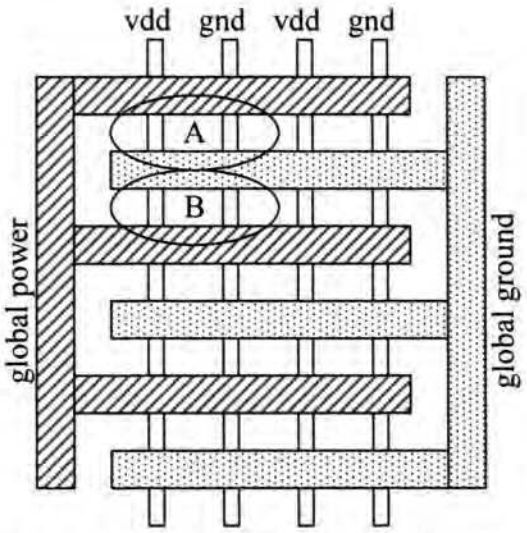


Figure 5.15 *Local and global power ground lines arrangement*

The top metal, Metal-6, is reserved for the power and ground lines linking every block of the internal core to the power ring lying around the chip. In blocks such as PPAs and VAs, the local power and ground lines, usually occupying Metal-1 and Metal-2, are arranged vertically along the columns of the compressors, such as "vdd" and "gnd" shown in Fig. 5.15. The global power and ground lines in Metal-6 are laid horizontally across the chip because the data bus lines are laid vertically. Thus, the stray capacitance between the data lines and the global power and ground are minimized. The horizontal interconnecting lines and the global power and ground lines are also carefully arranged to avoid overlapping. Both the horizontal global power lines and global ground lines are laid in

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

the shape of a pair of combs, with their fingers nearly clasped together as shown in Fig. 5.15. The comb structure of the global power line makes the local power (vdd) line segment enclosed in the ellipse "A" of Fig. 5.15 becomes parallelly connected with the local power line segment enclosed in the ellipse "B". Otherwise, they would have been connected in series. The same parallel connection is also implied in the local ground (gnd) line segments enclosed in the ellipses "A" and "B". Therefore, the current density requirement on the local power and ground lines is lowered and their widths can thus be minimized to reduce the area of the chip.

5.7 Design of the on-chip auxiliary circuit

There are a large number of inputs and outputs of the scalar product core. The number of input bits is $N\times R+N\times R$ = 16×16+16×16 = 512. The number of output bits is 36. It is impractical to use an IC package with more than 500 pins. Instead, a 32-pin package is used. The registers are used to multiplex the large number of inputs/outputs into the limited I/O pins. In order to eliminate the spurious computations during the register data preparation, each output of the register is gated by a D-latch, which is controlled by a signal "compute". Fig. 5.16 shows the structure of the input data registers. When "compute" is inactive, the D-latches retain the previous data fed into the scalar product core. The core is at rest and no extra power is dissipated before all the input data are written to the registers. When all data are registered, the signal "compute" is activated to release these input bits through the D-latches to the core for a new computation. Since "compute" triggers a new cycle of calculation, it will also be used in the delay detection circuits. The registers operate in two different modes. In the memory mode, each register can be accessed with a unique address. This mode is useful to configure the core as a normal vector multiplier. In the FIFO mode, the data can be moved from one register to the next. This mode allows the core to be used as a FIR filter.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

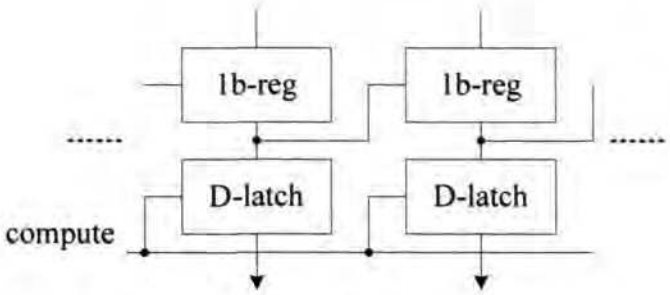


Figure 5.16 *Input registers*

To find out the worst case delay of a circuit, it is required to measure the delay of each computation. Several methods can be used, one of which is to measure every output to find out the elapsed time from the time the input is applied to the time the last stable output is obtained. The other is to feed the circuit with enough input data at high rate until the circuit fails to function [MAK96]. The period of the data rate at the verge of malfunction is the worst case delay. Both methods require reading the outputs at the correct time. Due to the pin limitation, not all the outputs of our design can be measured simultaneously, it is impossible to apply either method. A dedicated build-in circuit to measure the delay is ineluctable.

The delay detection circuit compares all the outputs of scalar product core with the expected outputs stored in the result compare registers. For our design, a row of 36 XNOR gates followed by a 36-input AND gate are used to compare the outputs and signify the completion of computation. The 36-input AND gate is implemented as a multi-level logic circuit as shown in Fig. 5.17. The structure is composed of seven 6-input AND gates and each of the 6-input AND gates is realized with a two-level NAND-NOR implementation comprising two 3-input NAND gates and one 2-input NOR gate. The extra delay of the XNOR gates and 36-input gate is introduced into the detection circuit. This delay surplus is not easy to measure on chip. Besides, it is also data dependent, i.e., the delay of the transition from the previous result Y_1 to the current result Y_2 is different from that of the transition from Y_3 to Y_2. Therefore, this surplus should be compensated from data to data, instead of treating it as a constant value.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

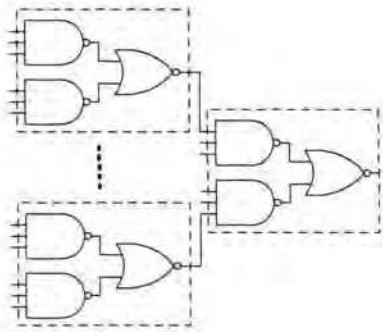


Figure 5.17 *The 36-input AND gate*

Fig. 5.18 shows the proposed delay detection circuit for 1 bit. It consists of two parts. One part enclosed in the dashed rectangle labeled "d" on the left is the core result comparison circuit, which generates an aggregate delay of the core and the detection circuit; the other part enclosed in the right dashed rectangle labeled "c" is the compensation circuit used to compensate for the detection circuit's delay. Prior to the computation when the signal "compute" is inactive, the expected result data should be set in the compare register. The multiplexer in the dashed rectangle "c" feeds the previously completed output Y_1, which comes from the output of the scalar product core, into the matching detection circuit, consisting of an XNOR gate followed by a 36-input AND gate. When a new computation is activated by the "compute" signal, the multiplexer of the compensate circuit switches from the core output to the expected result Y_2'. Thus the compensation circuit takes $t_{TG} + t_{XNOR} + t_{NAND36}$ to compare the change from Y_1 to Y_2' until the output "T_c" indicates a match. The core result comparison circuit in the dashed rectangle "d" is composed of an XNOR and a 36-input AND gates. The XNOR is constantly fed with the expected data Y_2' and the core's output. The latter is fed through a multiplexer on the left. The multiplexer connects both inputs to the core's output in order to insert a corresponding delay of t_{TG}. Therefore the core result comparison circuit takes $t_{CORE} + t_{TG} + t_{XNOR} + t_{NAND36}$ to generate a match indication in "T_d". The timing difference of "T_c" and "T_d", denoted as t_{CORE}, is the computational delay of the core. With this delay detection circuit, only two pins, "T_c" and "T_d", need to be monitored.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

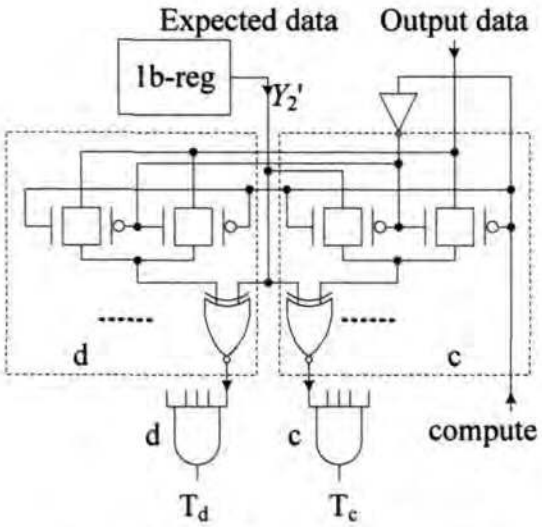


Figure 5.18 *Delay detection circuit*

The layout of the delay detection circuit is shown in Fig. 5.19. Every pair of 1-bit core result comparison circuit and compensation circuit should be placed as close as possible in the layout to minimize the wire length difference to the 36-input AND gates for a more accurate delay measurement.

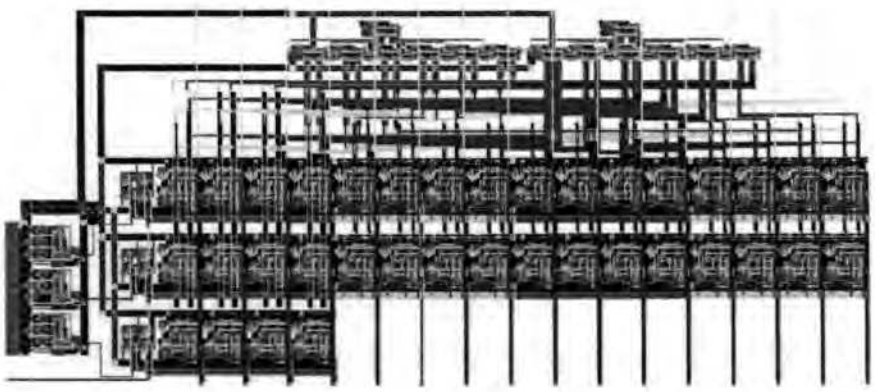


Figure 5.19 *Layout of delay detection circuit*

For the power measurement, only the power dissipation of the scalar product core is of interest. The power consumed by the auxiliary circuits should be excluded. We arranged two power supply pins, one for the core, the other for the auxiliary circuits, to facilitate the power measurement of the core.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

The layout of the chip completed with the auxiliary test circuitries has passed all design rules checks and verification tests. At the time of writing this thesis, it has just been submitted for a multi-project wafer (MPW) fabrication run.

5.8 Simulation Results

Three types of simulation of the two circuit architectures are made, namely the pre-layout circuit simulations of the conventional and the proposed architectures, and the post-layout simulation of the proposed circuit. The circuit simulator used is Synopsys Powermill Version 5.3. All the circuits are simulated at supply voltages range from 0.7V to 3.3V under the latest Chartered CSM 0.18μm CMOS technology. For pre-layout simulations, the architecture (high level structure) of the circuit is described in Verilog Hardware Description Language (Verilog HDL) and its gate level and transistor level (low level circuit) descriptions are written in SPICE format. For the post-layout simulation, the netlists, together with their parasitic parameters are extracted from the layout of the circuit. For fair comparison, both architectures are constructed from the same elementary cells and simulated with the same model file. The transistors models are from the CSM 0.18μm process library file revision 1G for Star-HSPICE Level 53. The resolution of the simulation time is set to 0.01ns. The frequency of computation (the rate at which data are input for simulation) is 50 MHz for supply voltages higher than 1V, and 10MHz for supply voltages below or equal to 1V. The 1024 input data patterns are generated randomly using MATLAB. All circuits are fed with the same input data stimuli. The average power dissipation at each voltage is determined from the measured power supply current averaged over the entire sequence of data inputs. The worst-case propagation delay is taken as the longest delay of the output from the input among all computations. The power efficiency is defined as the product of the average power dissipation and the worst case delay.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Table 5.4 *Comparison of power, delay, and power efficiency*

Voltage (V)		0.7	0.8	0.9	1.0	1.2	1.5	1.8	2.5	3.3
Data Rate (MHz)		10				50				
Power (mW)	**I**	1.968	2.410	2.947	3.542	25.96	44.33	72.07	180.1	402.6
	II	1.586	1.859	2.277	2.682	19.66	33.99	56.81	148.3	341.6
	III	1.845	2.152	2.574	3.144	23.39	39.48	64.96	166.1	379.8
Delay (ns)	**I**	68.17	40.63	28.74	20.85	13.11	8.949	7.226	5.080	4.202
	II	66.29	44.18	29.18	20.30	12.91	8.947	6.756	5.005	3.937
	III	72.82	44.21	29.70	20.62	13.33	10.00	6.919	5.162	4.259
Power efficiency (pJ)	**I**	134.2	97.92	84.70	73.85	340.3	396.7	520.8	914.9	1692
	II	105.1	82.13	66.44	54.44	253.8	304.1	383.8	742.2	1345
	III	134.3	95.14	76.45	64.83	311.7	394.8	449.5	857.4	1618

The final implementations of the conventional and proposed vector multiplier circuits use 189736 and 182990 transistors, respectively. The difference is due to the algorithmic and architectural improvement as both circuits use the same fundamental adder and compressor cells. The power dissipation, worst-case delay, and power efficiency at different voltages are measured and shown in Table 5.4. The pre-layout simulation results of the conventional circuit and the proposed circuit are denoted as I and II respectively, and the post-layout simulation results of the proposed circuit is denoted as III.

To highlight the differences between the pre- and post-layout results of our proposed circuit and its performance gain over the conventional implementation, the figures of merits of the three simulations are plotted against supply voltages in Fig. 5.20 to 5.22. Since power dissipation is proportional to the simulation frequency and the worst-case delay is unaffected by the data rate, to account for the different simulation frequencies used for the two supply voltage ranges, the average power and power efficiency are proportionally scaled down by 5 times when the supply voltage is higher than 1.0V.

The degradation in power dissipation and critical delay of Simulation III compared to those of Simulation II is expected because of the dominance of interconnect coupling capacitances and RC parasitic in advanced submicron technology. However, the severity of the degradation is far lower than anticipated due to the improved layout offered by our interconnect-centric design methodology. In fact, the post-layout circuit of our proposed

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

design has remarkably outperformed the pre-layout conventional circuit in terms of the power dissipation and power efficiency. The worst-case delay of our post-layout circuit is comparable to the pre-layout circuit of the conventional architecture, which has omitted the wire capacitances. Based on the layout experience of our proposed circuit, it would take probably more than 4 man-months to complete the full custom layout of the conventional design with irregular interconnections. Since the post-layout results of our actual circuit have already exceeded the pre-layout results of the conventional design, the physical layout of the conventional design is not conducted. If the parasitic parameters of the actual circuit of the conventional design are available, it is believed that the performance gain of our circuits will be more salient.

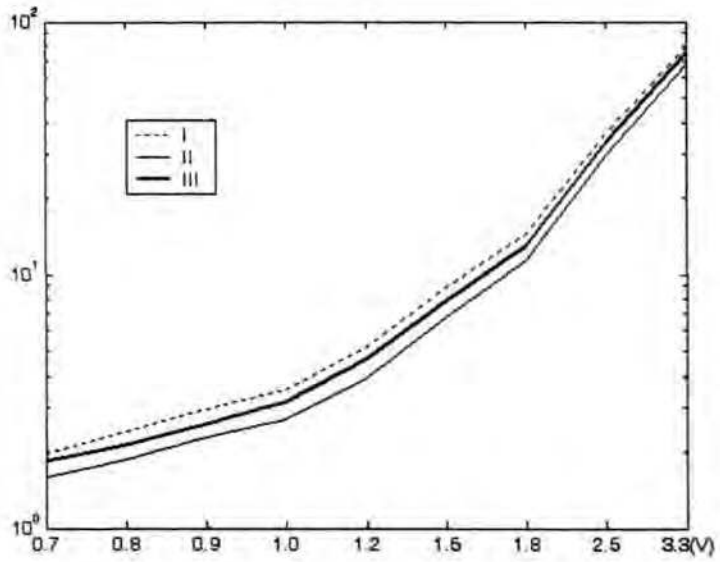


Figure 5.20 *Comparison of power dissipation* (mW/10MHz)

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

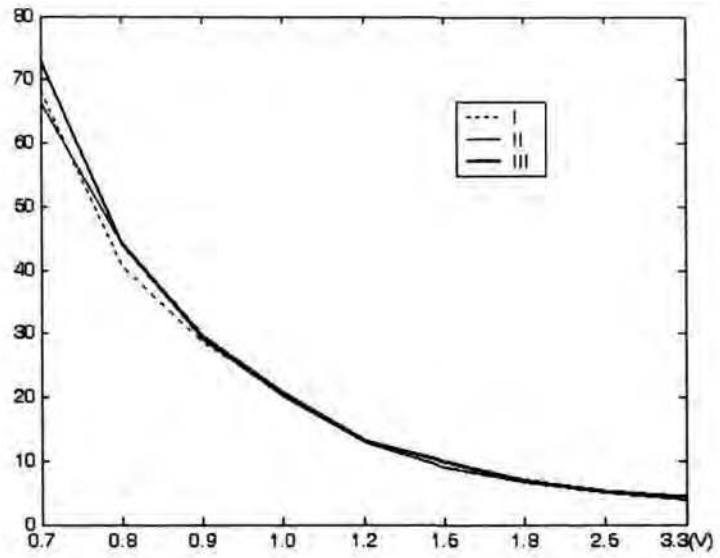


Figure 5.21 *Comparison of worst-case delay* (ns)

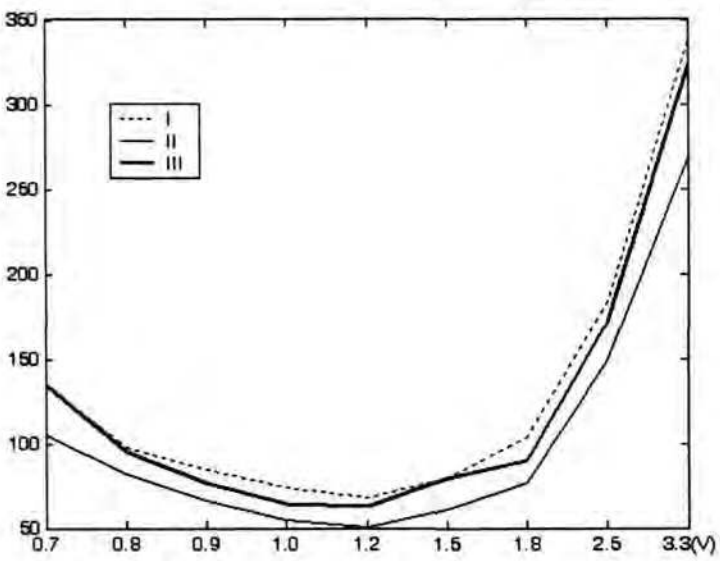


Figure 5.22 *Comparison of power efficiency* (pJ/10MHz)

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

5.9 Summary

A new algorithm for the design of a VLSI circuit for scalar product evaluation has been presented. The algorithm produced a novel full bit parallel architecture of scalar product macrocell featuring a low interconnect complexity, improved power efficiency and highly efficient VLSI area utilization. More importantly, the layout regularity and scalability enhance its performance superiority in deep submicron regime well above conventional VLSI design of vector processing unit for scalar product computation. The arithmetic core of the macrocell consists of a partial product generator, the partial product accumulator and the vector accumulator. Some auxiliary circuits including a unique delay detection circuit are proposed to enable accurate delay measurement of the core under the constraint of limited IO pins. The floor planning of the proposed architecture exploits the binary data locality through the border between the multiplication and accumulation operations based on a full combinational logic implementation. A comparison with the layout of the conventional vector multiplier shows that our proposed decomposition algorithm has led to a more compact, regular and modular physical design. A theoretical model for estimating the area and delay has been formulated. Comparing with the conventional architecture of the same capacity, the estimation shows that our design of a 16 bit scalar product multiplier on input vectors of 16 elements achieves a saving of 36.5% of silicon area, up to 59% increase in area usage efficiency and 23% decrease in interconnecting wire delay. The overall performances of average power consumption, worst case delay and power efficiency of our post-layout circuit surplus even the pre-layout circuit of conventional architecture when these circuits are simulated using Synopsys Nanosim over supply voltages from 0.7V to 3.3V based on Chartered CSM 0.18μm CMOS technology. The post-layout simulation shows that the worst case delay of the core at 1.8V is 6.92ns. At 50MHz input data rate, the power dissipation is 65.0mW. The relatively small deviations between the pre- and post-layout simulation results validate the inference of the theoretical estimation that the key contributors to the delay and power reduction of our proposed architecture are the shorter and balanced global and local interconnecting wires, a dominant factor in design consideration for VLSI circuits fabricated in the deep submicron technology.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Chapter 6

Covalent Redundant Binary Booth Encoded Multiplier

6.1 Introduction

Multiplication is widely used in digital signal processing applications, digital filtering for image, video and audio processing, and thousands of general purpose programs running in all architectures of computers. Therefore, digital multiplier is an obligatory and critical arithmetic unit in general purpose processors, digital signal processors and embedded systems. Its speed often determines how fast the processors can run.

A typical two's complement, also known as normal binary (NB) multiplier consists of four parts, namely the Booth encoder, the partial product generator, the partial product accumulator and the carry propagate adder (CPA). The modified Booth algorithm is adopted to reduce the number of partial products in order to speedup the partial product accumulation. The Booth encoder first recodes the multiplier to signed digit coefficients. The signed digit coefficients are then used to generate the partial products in the partial product generator. The partial products are essentially the integer multiples of the multiplicand, where some hard multiples that can not be realized by simple shifting and complementation operations need to be generated in advance by one or more carry propagate adders. The partial product accumulator, usually implemented in tree structure for high speed multiplier, sums up all the partial products into one carry stored number.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Finally, a CPA is used to convert the carry stored number to normal two's complement result.

The redundant binary (RB) number system was firstly introduced by Avizienis [AVI61] in 1961. Takagi [TAK85] proposed to apply this new arithmetic for fast multiplication and Edamatsu [EDA88] implemented it in VLSI. The RB system provides the propagation free addition for two RB numbers, making it a promising substitute for NB number system to implement the digital multiplier in the new DSM design paradigm. Besides, communications among redundant binary adders (RBA) within and across different layers of the RBA tree are simpler than those of the full and half adders of the carry save adder (CSA) tree of the NB multiplier. The use of RBA tree for the accumulation of partial products provides better optimization opportunity to highly modular and regular layout. The Booth encoding and the RB partial product generator affect the efficiency of the partial product generation, and the number of partial products that can be saved by this stage impacts the hardware cost and the RB multiplier structure. In this chapter, we scrutinize the existing methods for RB Booth encoder and partial product generator and propose some new algorithms for the design of low power high performance RB multipliers.

The chapter is organized as follows: Section 6.2 introduces the redundant binary systems followed by the structure of the RB multiplier, which can be decomposed into several constituent modules. Section 6.3 explores the existing algorithms and implementations of Booth encoders and partial product generators. An ingenious covalent redundant binary Booth encoding algorithm is proposed in Section 6.4. Its circuit implementations for different radix encoders and accompanying partial product generators are described in Section 6.5. A 54×54-bit multiplier is constructed based on the proposed covalent redundant binary Booth encoder and RB partial product generator, together with the existing RBA circuit for the RBA summing tree and RB-NB converter. The structure of the RB multiplier using the proposed algorithm is compared with the contender structures of other methods. Finally, the simulation results are analyzed and discussed in Section 6.6 and the chapter is closed with a summary in Section 6.7.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

6.2 Redundant binary coding and RB multiplier structures

A redundant binary (RB) number belongs to the signed digit number representation [AVI61], where the value of each digit is either "−1", "0", or "1". Therefore at least two binary bits are needed to represent each RB digit. According to the different mapping methods, there are three coding formats in RB number representation. One is the sign-magnitude coding shown in Table 6.1, where one bit, X^M, represents the magnitude "0" or "1" while the other, X^S, represents the sign, "+" or "−". This coding format can be expressed as (X^S, X^M). Negating a RB digit is equivalent to complementing the sign bit, i.e., $-(X^S, X^M) = (\overline{X^S}, X^M)$.

Table 6.1 *Sign-magnitude coding*

Coding (X^S, X^M)	RB digit d
(0, 0)	0
(0, 1)	1
(1, 0)	0
(1, 1)	$\bar{1}$

The second format is the positive-negative coding. The relationship of the represented value d and its coding bits X^+, X^- is given in (6.1).

$$d = X^+ - X^- \tag{6.1}$$

The positive-negative coding is popularly used by recent published papers due to the ease of its generation from the normal binary numbers. Table 6.2 shows the positive-negative coding in (X^+, X^-). Negating a digit can be simply obtained by exchanging the two coding bits, i.e., $-(X^+, X^-) = (X^-, X^+)$.

Table 6.2 *Positive-negative coding*

Coding (X^+, X^-)	RB digit d
(0, 0)	0
(0, 1)	$\bar{1}$
(1, 0)	1
(1, 1)	0

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

The remaining format is the positive-negative complement coding. The relationship between the value of the digit and its coding bits is given by (6.2)

$$d = X^+ - \overline{X^-} \tag{6.2}$$

Table 6.3 shows the signed digits represented by this coding. Negating a RB digit is equivalent to complementing both coding bits, i.e., $-(X^+, X^-) = (\overline{X^+}, \overline{X^-})$

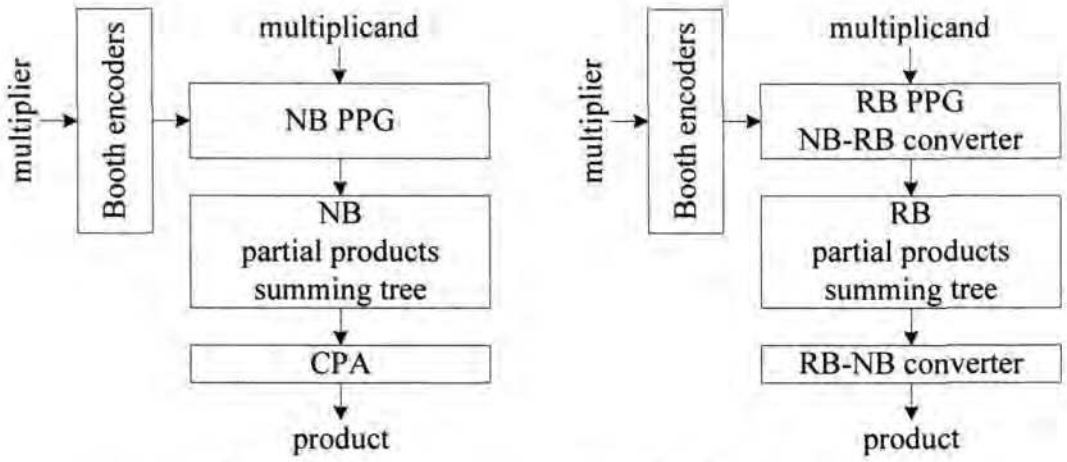
Table 6.3 *Positive-negative complement coding*

Coding (X^+, X^-)	**RB digit** d
(0, 0)	$\bar{1}$
(0, 1)	0
(1, 0)	0
(1, 1)	1

Since positive-negative coding is the most frequently used coding format for RB multiplication, the multipliers and their algorithms and architectures discussed in this chapter are designed based on this coding format.

As the peripheral interfaces of most digital systems are based on the normal binary number system, the RB multiplier needs to convert the normal binary multiplicand to redundant binary number. After obtaining the product in the redundant binary format, it is necessary to convert it back to the normal binary format. Therefore a RB Booth's multiplier is composed of four parts, which are the Booth encoder to generate the signed digit coefficients for the RB multiples, the RB partial product generator to convert the NB multiplicand to RB partial products according to the signed digit coefficients, the RB partial product summing tree to compress the partial products to one RB result, and the RB to NB converter to output the result in normal binary format for standard bus interface. The structure of the RB multiplier is shown in Fig. 6.1(b). For comparison, the structure of the NB multiplier is shown in Fig. 6.1(a).

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library



(a) *normal binary multiplier* (b) *redundant binary multiplier*

Figure 6.1 *Structure of Booth digital multiplier*

Although the RB summing tree is in general more costly to implement, the Booth encoder and the RB partial product generator determine how efficient these RB partial products are generated, and therefore indirectly dictate the amount of hardware needed to construct the multiplier. Before explaining our proposed algorithm, it is necessary to explore the existing methods for Booth encoding and RB partial product generation.

6.3 Existing Booth encoding and partial product generation methods for RB multiplication

The Booth algorithm [BOO51] is an efficient way to reduce the number of partial products in fast signed digital multiplier because it groups any series of "1"s in one of the operands called the multiplier (the other operand is called the multiplicand). The modified Booth encoding algorithm [MAC61] is a parallel counterpart of the serial Booth encoding. Therefore, it is more suitable for the design and implementation of high-speed digital multiplier in hardware. For convenience, we refer to the modified Booth encoding as Booth encoding. As the radix value, 2^N (N = 1, 2, 3...) of the Booth-N encoding increases, the number of recoded Booth digits for the multiplier also decreases by a factor to $1/N$, so does the number of partial products.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

6.3.1 Normal Booth encoding

In Booth-N encoding, a Booth-N digit d_i is equivalent to N normal binary bits. Besides the N bits to be encoded, the generation of a Booth-N digit requires one additional bit to be borrowed from the bit to the right of the LSB of this N-bit string. If $b_{i+N-1}\ b_{i+N-2} \ldots b_{i+1}\ b_i$ are the multiplier bits to be encoded, then b_{i-1} will be the borrowed bit, which is also called the overlapping bit. The encoded digit has the value given by (6.3).

$$d_i = -2^{N-1} b_{i+N-1} + \sum_{n=0}^{N-2} 2^n b_{i+n} + b_{i-1} \tag{6.3}$$

Tables 6.4 to 6.7 show the Booth encoded digits and their corresponding binary bits with the overlapping bit in bracket. The multiples represented by the Booth encoded digits are also shown in the tables, where M denotes the multiplicand.

When N = 1, the Booth-1 digit d_i is converted from $b_i(b_{i-1})$. The encoded digit has the value given by:

$$d_i = -b_i + b_{i-1} \tag{6.4}$$

The possible digits are $\bar{1}$, 0 and 1. Their mapping is listed in Table 6.4.

Table 6.4 *Booth-1 encoding*

Normal Binary $b_i\ (b_{i-1})$	**Booth-1 Digit** d_i	**Multiple** xM
0 0	0	0
0 1	1	$1M$
1 0	$\bar{1}$	$-1M$
1 1	0	0

Booth-1 encoding is seldom used in normal Booth multiplier because it does not help to reduce the number of partial products compared with non-Booth encoded multiplier.

In Booth-2 encoding, N = 2, and every Booth-2 digit, D_i is mapped from the bits $b_{i+1}b_i(b_{i-1})$. Therefore, the value of the encoded digit D_i is given by:

$$D_i = -2b_{i+1} + b_i + b_{i-1} \tag{6.5}$$

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

The legitimate digits in Booth-2 encoded number are 2, 1, 0, $\bar{1}$ and $\bar{2}$. The mapping from the normal binary bits to the Booth-1 digits and the Booth-2 digit are shown in Table 6.5.

Table 6.5 *Booth-2 encoding*

Normal Binary $b_{i+1}\ b_i\ (b_{i-1})$	**Booth-1 Digits** $d_{i+1}\ d_i$	**Booth-2 Digit** D_i	**Multiple** xM
0 0 0	0 0	0	0
0 0 1	0 1	1	$1M$
0 1 0	1 $\bar{1}$	1	$1M$
0 1 1	1 0	2	$2M$
1 0 0	$\bar{1}$ 0	$\bar{2}$	$-2M$
1 0 1	$\bar{1}$ 1	$\bar{1}$	$-1M$
1 1 0	0 $\bar{1}$	$\bar{1}$	$-1M$
1 1 1	0 0	0	0

In Booth-3 and Booth-4 encoding, the values of the encoded digits are expressed by (6.6) and (6.7), respectively.

$$d_i = -2^2 b_{i+2} + 2b_{i+1} + b_i + b_{i-1} \tag{6.6}$$

$$d_i = -2^3 b_{i+3} + 2^2 b_{i+2} + 2b_{i+1} + b_i + b_{i-1} \tag{6.7}$$

Tables 6.6 and 6.7 show the normal binary bits and their equivalent multiples for Booth-3 and Booth-4 encoding, respectively, where the hard multiples are marked with "*".

Table 6.6 *Booth-3 encoding*

Normal Binary $b_{i+2}\ b_{i+1}\ b_i\ (b_{i-1})$	**Multiple** xM	**Normal Binary** $b_{i+2}\ b_{i+1}\ b_i\ (b_{i-1})$	**Multiple** xM
0 0 0 (0)	0	1 0 0 (0)	$-4M$
0 0 0 (1)	M	1 0 0 (1)	$-3M$*
0 0 1 (0)	M	1 0 1 (0)	$-3M$*
0 0 1 (1)	$2M$	1 0 1 (1)	$-2M$
0 1 0 (0)	$2M$	1 1 0 (0)	$-2M$
0 1 0 (1)	$3M$*	1 1 0 (1)	$-M$
0 1 1 (0)	$3M$*	1 1 1 (0)	$-M$
0 1 1 (1)	$4M$	1 1 1 (1)	-0

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Table 6.7 *Booth-4 encoding*

Normal Binary $b_{i+3}\ b_{i+2}\ b_{i+1}\ b_i\ (b_{i-1})$	**Multiple** xM	**Normal Binary** $b_{i+3}\ b_{i+2}\ b_{i+1}\ b_i\ (b_{i-1})$	**Multiple** xM
0 0 0 0 (0)	0	1 0 0 0 (0)	$-8M$
0 0 0 0 (1)	M	1 0 0 0 (1)	$-7M$*
0 0 0 1 (0)	M	1 0 0 1 (0)	$-7M$*
0 0 0 1 (1)	$2M$	1 0 0 1 (1)	$-6M$*
0 0 1 0 (0)	$2M$	1 0 1 0 (0)	$-6M$*
0 0 1 0 (1)	$3M$*	1 0 1 0 (1)	$-5M$*
0 0 1 1 (0)	$3M$*	1 0 1 1 (0)	$-5M$*
0 0 1 1 (1)	$4M$	1 0 1 1 (1)	$-4M$
0 1 0 0 (0)	$4M$	1 1 0 0 (0)	$-4M$
0 1 0 0 (1)	$5M$*	1 1 0 0 (1)	$-3M$*
0 1 0 1 (0)	$5M$*	1 1 0 1 (0)	$-3M$*
0 1 0 1 (1)	$6M$*	1 1 0 1 (1)	$-2M$
0 1 1 0 (0)	$6M$*	1 1 1 0 (0)	$-2M$
0 1 1 0 (1)	$7M$*	1 1 1 0 (1)	$-M$
0 1 1 1 (0)	$7M$*	1 1 1 1 (0)	$-M$
0 1 1 1 (1)	$8M$	1 1 1 1 (1)	-0

Intuitively, selecting a higher radix Booth algorithm to encode the multiplier would reduce more partial products for a faster multiplier to be constructed. However, a close examination of Tables 6.6 and 6.7 will reveal that the number of multiples increases commensurately with the radix to 2^N+1. Besides, the number of hard multiples which are not the powers of two also increases. For example, in Booth-3 encoding, there are two hard multiples, $3M$ and $-3M$ out of a total of nine distinct multiples while in Booth-4 encoding, there are eight hard multiples, which are $\pm 3M$, $\pm 5M$, $\pm 6M$, $\pm 7M$ out of seventeen distinct multiples. All these hard multiples cannot be obtained by simple shifting and/or complementation operations on the multiplicand. Additional time consuming carry propagate adders are required to generate them before the partial products are generated. The advantage of Booth-3 and higher radix Booth encoding is compromised due to the long delay and complex decoding logic required for the generation of the hard multiples.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

6.3.2 NBE-RBPPG: RB partial product generation with normal Booth encoding

An RB partial product, R can be obtained by two NB partial products, A and B with the definition of the positive-negative coding format. This method of generating RB partial products is abbreviated as NBE-RBPPG.

$$R = A + B = A - (-B) \tag{6.8}$$

Since $-B = \overline{B} + 1$, substituting it into (6.8) gives:

$$R = A - (\overline{B} + 1) = A - \overline{B} - 1 \tag{6.9}$$

If A and B are K-bit two's complement numbers, then

$$A = -2^{K-1} a_{K-1} + \sum_{i=0}^{K-2} 2^i a_i \tag{6.10}$$

and

$$B = -2^{K-1} b_{K-1} + \sum_{i=0}^{K-2} 2^i b_i , \tag{6.11}$$

The number of RB digits of the RB partial product is also K.

$$\begin{aligned} R &= \left(-2^{K-1} a_{K-1} + \sum_{i=0}^{K-2} 2^i a_i \right) - \left(-2^{K-1} \overline{b_{K-1}} + \sum_{i=0}^{K-2} 2^i \overline{b_i} \right) - 1 \\ &= 2^{K-1} \left(\overline{b_{K-1}} - a_{K-1} \right) + \sum_{i=0}^{K-2} 2^i \left(a_i - \overline{b_i} \right) - 1 \end{aligned} \tag{6.12}$$

It should be noted that the most significant digit has been negated. The RB digit string of the partial product, R expressed in positive-negative coding format is:

$$(\overline{b_{K-1}}, a_{K-1})(a_{K-1}, \overline{b_{K-1}}) \cdots (a_1, \overline{b_1})(a_0, \overline{b_0}) \quad + \quad (0,1) \tag{6.13}$$

If A and B are unsigned numbers used as the mantissa of the floating point number in compliance with the IEEE 754 standard, then the RB partial product is

$$R = \left(\sum_{i=0}^{K-1} 2^i a_i \right) - \left(\sum_{i=0}^{K-1} 2^i \overline{b_i} \right) - 1 = \sum_{i=0}^{K-2} 2^i \left(a_i - \overline{b_i} \right) - 1 \tag{6.14}$$

The RB number, expressed in positive-negative format is

$$(a_{K-1}, \overline{b_{K-1}})(a_{K-1}, \overline{b_{K-1}}) \cdots (a_1, \overline{b_1})(a_0, \overline{b_0}) \quad + \quad (0,1) \tag{6.15}$$

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

[MAK96] used this method and the normal Booth encoder to implement a 54×54-bit RB multiplier. The two NB partial products generated by two adjacent Booth encoders compose one RB partial product. Due to the limitation of the hard multiples, only up to Booth-2 encoding are used. The algorithm is straightforward, but every RB partial products requires one constant RB digit (0,1) to correct its value, as indicated in (6.13) and (6.15). All the correction constants of the RB partial products can be accumulated to form a new RB partial product, called the RB correction vector. For the Booth-N multiplier of word length K, the number of RB partial products generated is:

$$\left\lceil \frac{\lceil K/N \rceil}{2} \right\rceil + 1 .$$

The RB correction vector incurs additional hardware for its accumulation. It can even increase the number of stages of the summing tree, if the word length of the multiplier is 2^n, such as the multipliers in single extended and double extended floating point number formats, whose effective mantissa are 32 and 64, respectively. Consequently, the power dissipation and worst case delay are also degraded by the inclusion of the correction vector.

Fig. 6.2 shows a pair of normal Booth-2 encoders used to generate the coefficients of the RB partial products. As the encoder outputs the sign and magnitude signals, the NB partial products are generated from the non-negative multiples of the multiplicand. To obtain the negative NB multiples, the positive multiples need to be complemented followed by an addition of 1. Therefore, it is necessary to send both sign signals of the Booth encoders to correct the RB partial product.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

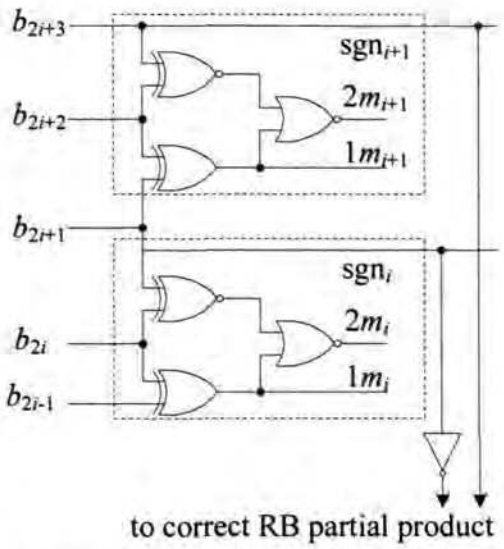


Figure 6.2 *NBE-RBPPG Booth encoder*

Fig. 6.3 shows one digit of the RB partial product generator, which is composed of two one bit NB partial product generators.

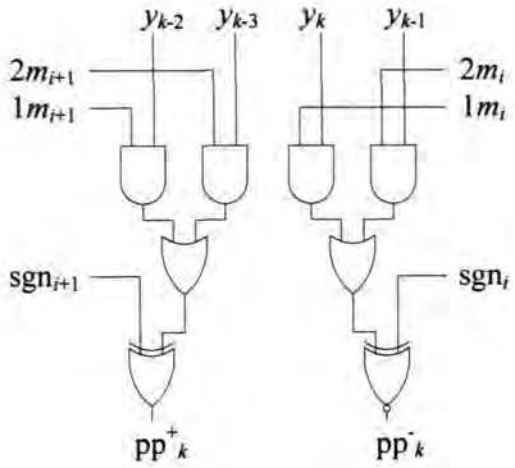


Figure 6.3 *One digit redundant binary partial product generator for NBE-RBPPG*

6.3.3 RBSD Booth encoding and partial product generation

To address the problem of generating hard multiples in high-radix Booth encoding, Besli and Deshmukh [BES02(1), BES02(2)] noticed that some multiples can be obtained by subtracting one non-hard multiple from another. The partial products generated in this

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

manner fit the format of the positive-negative RB coding. This is realized by a different Booth encoding logic, called the RB Signed Digit (RBSD) Booth encoding.

Table 6.8 shows the RBSD Booth-3 encoding, where the original hard multiples $\pm 3M$ are replaced by $\pm(4M - M)$. Table 6.9 shows the RBSD Booth-4 encoding.

Table 6.8 *RBSD Booth-3 encoding*

Normal Binary	**Multiple**		**Normal Binary**	**Multiple**	
$b_{i+2}\ b_{i+1}\ b_i\ (b_{i-1})$	$+M$	$-M$	$b_{i+2}\ b_{i+1}\ b_i\ (b_{i-1})$	$+M$	$-M$
0 0 0 (0)	0	0	1 0 0 (0)	0	$4M$
0 0 0 (1)	M	0	1 0 0 (1)	M	$4M$
0 0 1 (0)	M	0	1 0 1 (0)	M	$4M$
0 0 1 (1)	$2M$	0	1 0 1 (1)	0	$2M$
0 1 0 (0)	$2M$	0	1 1 0 (0)	0	$2M$
0 1 0 (1)	$4M$	M	1 1 0 (1)	0	M
0 1 1 (0)	$4M$	M	1 1 1 (0)	0	M
0 1 1 (1)	$4M$	0	1 1 1 (1)	0	0

Table 6.9 *RBSD Booth-4 encoding*

Normal Binary	**Multiple**		**Normal Binary**	**Multiple**	
$b_{i+3}\ b_{i+2}\ b_{i+1}\ b_i\ (b_{i-1})$	$+M$	$-M$	$b_{i+3}\ b_{i+2}\ b_{i+1}\ b_i\ (b_{i-1})$	$+M$	$-M$
0 0 0 0 (0)	0	0	1 0 0 0 (0)	0	$8M$
0 0 0 0 (1)	M	0	1 0 0 0 (1)	M	$8M$
0 0 0 1 (0)	M	0	1 0 0 1 (0)	M	$8M$
0 0 0 1 (1)	$2M$	0	1 0 0 1 (1)	$2M$	$8M$
0 0 1 0 (0)	$2M$	0	1 0 1 0 (0)	$2M$	$8M$
0 0 1 0 (1)	$4M$	M	1 0 1 0 (1)	0	$5M$*
0 0 1 1 (0)	$4M$	M	1 0 1 1 (0)	0	$5M$*
0 0 1 1 (1)	$4M$	0	1 0 1 1 (1)	0	$4M$
0 1 0 0 (0)	$4M$	0	1 1 0 0 (0)	0	$4M$
0 1 0 0 (1)	$5M$*	0	1 1 0 0 (1)	M	$4M$
0 1 0 1 (0)	$5M$*	0	1 1 0 1 (0)	M	$4M$
0 1 0 1 (1)	$8M$	$2M$	1 1 0 1 (1)	0	$2M$
0 1 1 0 (0)	$8M$	$2M$	1 1 1 0 (0)	0	$2M$
0 1 1 0 (1)	$8M$	M	1 1 1 0 (1)	0	M
0 1 1 1 (0)	$8M$	M	1 1 1 1 (0)	0	M
0 1 1 1 (1)	$8M$	0	1 1 1 1 (1)	0	0

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Among the four hard multiples in the original Booth-4 encoding, $3M$, $6M$ and $7M$ are easily obtained by the combination of two non-hard multiples in positive-negative pair. The only exception is the hard multiple $5M$ (marked by "*" in Table 6.9), which cannot be generated in this manner. Therefore, additional hardware is necessary to generate the $5M$ multiple. A simple RB adder can be used to form the $5M$ multiple by adding $4M$ and M as shown in Fig. 6.4 [BES02(2)]. Fortunately, this RB addition is carry-free and it does not lie in the critical path of the RBSD Booth encoder and RB partial product generator circuit.

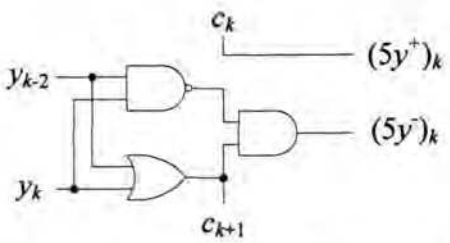


Figure 6.4 *RB adder for generating the k-th RB digit of the 5M hard multiple*

Fig. 6.5 to 6.7 show the RBSD Booth-2 to Booth-4 encoders and the RB partial product (RBPP) generators, where pp^+_k and pp^-_k are the two coding bits of the k-th RB digit of the partial product.

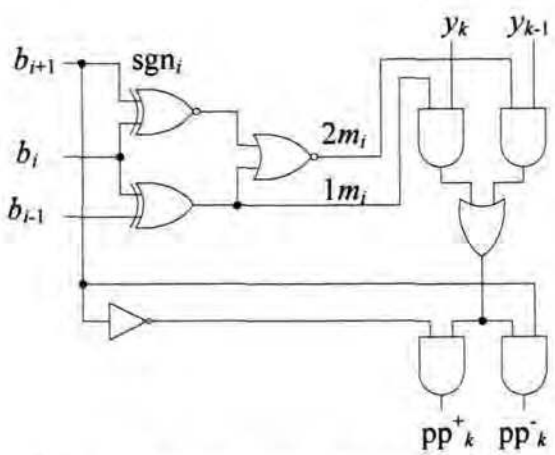


Figure 6.5 *RBSD Booth-2 encoder and RBPP generator*

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

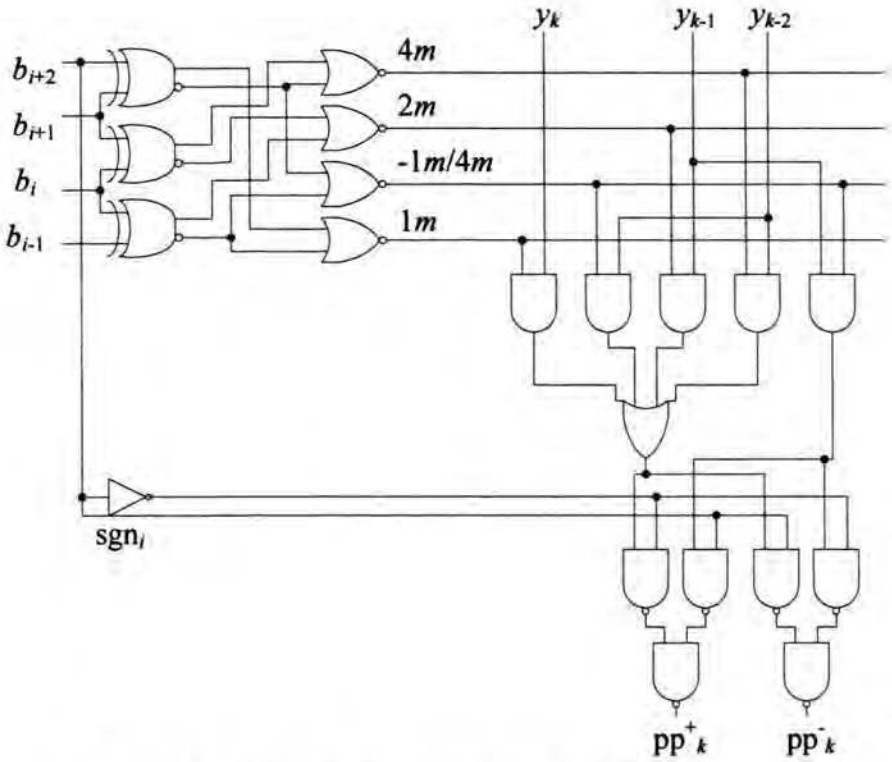


Figure 6.6 *RBSD Booth-3 encoder and RBPP generator*

It is obvious that the RBSD Booth-3 and Booth-4 encoders are much more complex, which use high fan-in gates, especially in the partial product generator circuit. Since the circuit for each digit of the RB partial product will be duplicated in a large number, more hardware is incurred.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

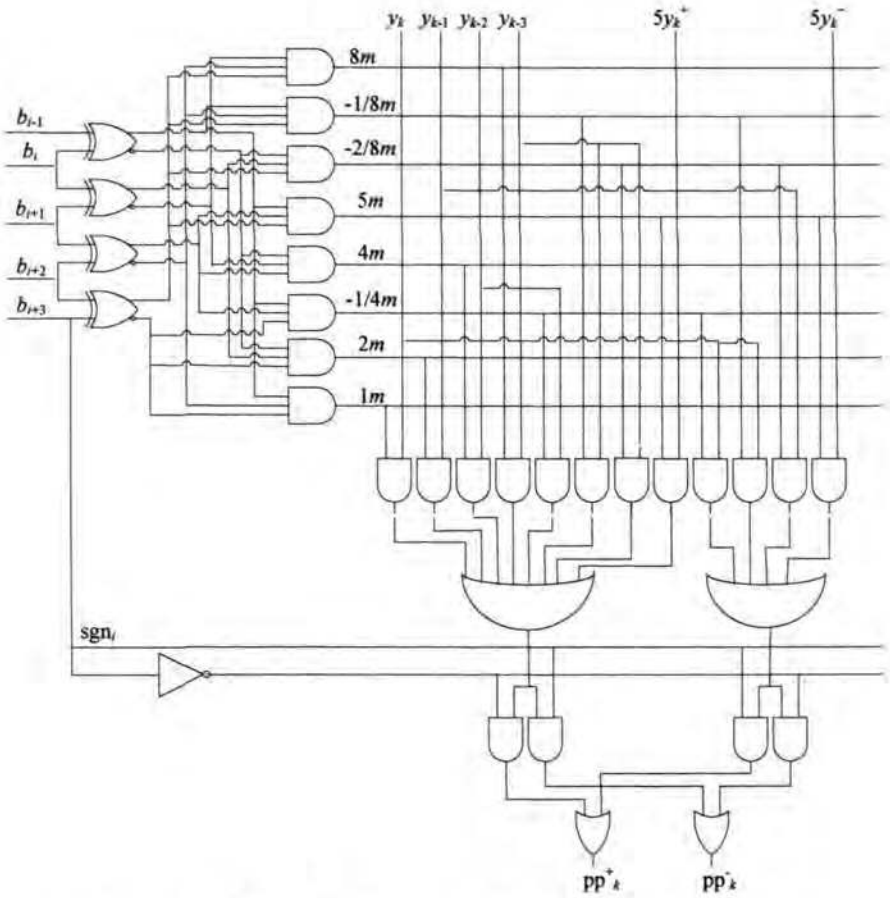


Figure 6.7 *RBSD Booth-4 encoder and RBPP generator*

On the first face, it appears that the RBSD encoding is a "perfect" solution as there is no correction vector needed for the RB partial products, and the area and time consuming hardware for generating the hard multiples in high radix Booth encoding have been eliminated. However, the cost of achieving the above merits is an increase (almost doubling) in the number of RB partial products. This is because only one Booth encoded coefficient is used for generating a RB partial product as opposed to the earlier method where two Booth encoded coefficients can be exhausted to generate one RB partial product. Half of the binary bits representing the RB partial product generated from the non-hard multiple in the RBSD encoding are filled with "0"s, which is very inefficient. For the RBSD Booth-N multiplier of word length K, the total number of RB partial products generated is:

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

$$\left\lceil \frac{K}{N} \right\rceil,$$

which is almost twice that of the previous NBE method. One extra layer of RB partial product summing tree is needed, which increases both the power dissipation and the worst case delay.

6.4 CRBBE: Proposed Booth encoding and partial product generation algorithms for RB multiplication

In this section, we propose a novel Covalent Redundant Binary Booth Encoding (CRBBE) algorithm to overcome the shortcoming of the RBSD Booth encoding method presented in the last section. Our algorithm utilizes the characteristics of the Booth encoded numbers to generate a reduced number of RB partial products without inducing any RB correction vector.

6.4.1 CRBBE-1: the Booth-1 encoding for RB multiplication

From (6.4), any pair of contiguous encoded digits $d_{i+1}d_i$ of Booth-1 encoding, which is mapped from two adjacent bits of the normal binary number $b_{i+1}(b_{(i+1)-1})$ and $b_i(b_{i-1})$ (refer to Table 6.4 for the mapping) can never exhaust all possible combinations of two digits from the set $\{0, 1, \bar{1}\}$. This is because of the constraint that the borrowed bit from which the digit d_{i+1} is mapped must come from the MSB of the binary bits from which its right contiguous digit d_i is mapped, i.e. $b_{(i+1)-1} \equiv b_i$. Table 6.10 shows the permissible combinations of contiguous digit pairs in Booth-1 encoded number. They are grouped into four categories according to the left digit d_{i+1}.

Table 6.10 *Permissible pairs of contiguous digits (d_{i+1} d_i) in Booth-1 encoded number*

$d_{i+1}=1$	$d_{i+1}=0$	$d_{i+1}=\bar{0}$	$d_{i+1}=\bar{1}$
$1\bar{0}$	01	$\bar{0}\bar{0}$	$\bar{1}1$
$1\bar{1}$	00	$\bar{0}1$	$\bar{1}0$

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

In a redundant binary multiplier, a RB partial product can be formed straightforwardly by grouping the same weighted bits from two normal binary partial products generated from a pair of adjacent Booth encoders using any of the coding formats presented in Section 6.2. Thus, if two adjacent normal binary Booth encoders always generate signed digit coefficients of opposite signs, their corresponding NB partial products can easily be combined to form a single positive-negative coded RB partial product without any correction digit. This is possible if the contiguous digits of the Booth encoded multiplier alternate in signs. We call this algorithm to generate a compound RB partial product from two Booth encoded digits the Covalent Redundant Binary Booth encoding (CRBBE) for its analogy to the way a covalent compound is formed from charge sharing. From Table 6.10, the criterion of having opposite polarity contiguous encoded digits can be fulfilled by rewriting the duplets such that one digit of the pair is positive and the other digit is negative without changing the value of the compound multiple, i.e., $2\,d_{i+1} + d_i$ remains unchanged. Some duplets in the columns $d_{i+1}=0$ and $d_{i+1}=\bar{0}$ of Table 6.10 need to be rewritten. Using the fact that the digit 0 is neutral and is equivalent to $\bar{0}$. We can reformat the duplets 01 to $\bar{0}\,1$, 00 to $\bar{0}\,0$, $\bar{0}\bar{0}$ to $0\,\bar{0}$, and $\bar{0}\bar{1}$ to $0\,\bar{1}$ to obtain the required positive-negative coefficient pair without changing the values of their compound multiples. Table 6.11 shows the resultant contiguous digit pairs for the proposed Covalent RB Booth-1 (CRBBE-1) encoding.

Table 6.11 *Proposed Covalent RB Booth-1 encoded duplet ($d_{i+1}\,d_i$)*

$d_{i+1}=1$	$d_{i+1}=0$	$d_{i+1}=\bar{0}$	$d_{i+1}=\bar{1}$
$1\,\bar{0}$ = (2, 0) $1\,\bar{1}$ = (2, 1)	$01=\bar{0}\,1$=(1, 0) $00=\bar{0}\,0$=(0, 0)	$\bar{0}\bar{0}=0\,\bar{0}$=(0, 0) $\bar{0}\bar{1}=0\,\bar{1}$=(0, 1)	$\bar{1}\,1$ = (1, 2) $\bar{1}\,0$ = (0, 2)

In Table 6.11, the compound signed digit coefficient, $r_i = (r^+_i, r^-_i) = r^+_i - r^-_i$ generated by each duplet, $d_{i+1}d_i$ is also shown in bracket. The shaded cells in Table 6.11 represent the positive-negative pair while the unshaded cells represent the negative-positive pair. For every positive-negative digit pair in the shaded cells, there will be a corresponding negative-positive digit pair in the unshaded cells with their resultant multiples differ only

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

in sign. This property can be used to greatly simplify the hardware for the CRBBE circuit so that only the absolute values of the coefficients need to be generated.

Thus, the values of r^+ and r^- are non-negative as obtained from the outputs of the adjacent Booth encoders. Negative compound coefficient can be obtained from its positive counterpart by simply swapping the values of r^+ and r^-. To simplify the design of the Booth encoder, the signs of the digits d_{i+1} and d_i are ignored at the outset. The sign of the least significant digit d_i in the duplet is then used to determine if the swapping of the coefficients r^+_i and r^-_i is necessary. The criterion for the ordering of the coefficients is given by (6.16):

$$(r_i^+, r_i^-) = \begin{cases} (2|d_{i+1}|, |d_i|) & \text{if } d_i \text{ is negative} \\ (|d_i|, 2|d_{i+1}|) & \text{if } d_i \text{ is positive} \end{cases} \tag{6.16}$$

6.4.2 CRBBE-2: the Booth-2 encoding for RB multiplication

Based on the similar rationales as in the case of the Booth-1 encoding, two contiguous digits, $D_{i+1}D_i$, of Booth-2 encoding mapping from three contiguous bits $b_{2(i+1)+1}$ $b_{2(i+1)}$ ($b_{2(i+1)-1}$) and b_{2i+1} b_{2i} (b_{2i-1}) of the normal binary multiplier have also some restrictions on its legal combinations, which are shown in Table 6.12.

Table 6.12 *Permissible pairs of contiguous digits (D_{i+1} D_i) in Booth-2 encoded number*

D_{i+1}=2	D_{i+1}=1	D_{i+1}=0	D_{i+1}=$\bar{0}$	D_{i+1}=$\bar{1}$	D_{i+1}=$\bar{2}$
$2\bar{0}$	12	02	$\bar{0}\bar{0}$	$\bar{1}2$	$\bar{2}2$
$2\bar{1}$	11	01	$\bar{0}\bar{1}$	$\bar{1}1$	$\bar{2}1$
$2\bar{2}$	10	00	$\bar{0}2$	$\bar{1}0$	$\bar{2}0$
	$1\bar{0}$			$\bar{1}\bar{0}$	
	$1\bar{1}$			$\bar{1}\bar{1}$	
	$1\bar{2}$			$\bar{1}\bar{2}$	

Most digit pairs in Table 6.12 have a positive-negative or negative-positive format. Those that are not can be easily changed to one of these formats, except for the duplets 12, 11, and 10 in the column D_{i+1}=1, and the duplets $\bar{1}\bar{0}$, $\bar{1}\bar{1}$ and $\bar{1}\bar{2}$ in the column D_{i+1}=$\bar{1}$. The

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

duplets 12 and $\bar{1}\bar{2}$ can be rewritten as $2\,\bar{2}$ and $\bar{2}\,2$, respectively to form the required format while maintaining the same compound coefficient value. Similarly, the duplets 10 and $\bar{1}0$ can be rewritten to $1\,\bar{0}$ and $\bar{1}\,0$, respectively. However, the duplets 11 and $\bar{1}\bar{1}$, which represent the compound coefficients for the RB multiples of $5M$ and $-5M$, respectively, cannot be easily reformatted to any of the desirable positive-negative pairs. To facilitate the high speed partial product generation in hardware design, the hard RB multiple of $5M$ needs to be calculated in advance by dedicated circuit.

The duplets in columns $D_{i+1}=0$ and $D_{i+1}=\bar{0}$ do not have the desirable positive-negative format either. These cases are trivial as the required format can be obtained by replacing 0 with $\bar{0}$ and $\bar{0}$ with 0 in the D_{i+1} position. It does not make the circuit design more complex if we treat the outputs of the Booth encoders as the absolute value of the compound coefficient (r^+_i, r^-_i) and use the sign of D_{i+1} to determine if it belongs to the positive-negative or negative-positive format similar to what have been done for CRBBE-1 encoding. Table 6.13 shows the reformatted duplets for the proposed Covalent Redundant Binary Booth-2 Encoding (CRBBE-2). The duplets 11 and $\bar{1}\bar{1}$ correspond to the two hard multiples are marked by "*" to indicate that they need to be preprocessed independently.

Table 6.13 *Proposed Covalent RB Booth-2 encoded duplets ($D_{i+1}D_i$)*

$D_{i+1}=2$	$D_{i+1}=1$	$D_{i+1}=0$	$D_{i+1}=\bar{0}$	$D_{i+1}=\bar{1}$	$D_{i+1}=\bar{2}$
$2\,\bar{0}$	$12=2\,\bar{2}$	$02=\bar{0}\,2$	$\bar{0}\bar{0}=0\,\bar{0}$	$\bar{1}\,2$	$\bar{2}\,2$
$2\,\bar{1}$	11^*	$01=\bar{0}\,1$	$\bar{0}\,1=0\,\bar{1}$	$\bar{1}\,1$	$\bar{2}\,1$
$2\,\bar{2}$	$10=1\,\bar{0}$	$00=\bar{0}\,0$	$\bar{0}\bar{2}=0\,\bar{2}$	$\bar{1}\,0$	$\bar{2}\,0$
	$1\,\bar{0}$			$\bar{1}\bar{0}=\bar{1}\,0$	
	$1\,\bar{1}$			$\bar{1}\bar{1}^*$	
	$1\,\bar{2}$			$\bar{1}\bar{2}=\bar{2}\,2$	

From Table 6.13, the coefficient reordering criterion is given by (6.17):

$$(r_i^+, r_i^-) = \begin{cases} (4|D_{i+1}|, |D_i|) & \text{if } D_{i+1} \text{ is positive, except } 11 \\ (|D_i|, 4|D_{i+1}|) & \text{if } D_{i+1} \text{ is negative, except } \bar{1}\bar{1} \end{cases} \tag{6.17}$$

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

6.4.3 CRBBE-1.5: A mixture of CRBBE-1 and CRBBE-2

The proposed CRBBE-1 and CRBBE-2 algorithms use either two Booth-1 or two Booth-2 encoders to generate one compound RB partial product. To avoid using the hardware to generate the hard RB multiples of $5M$ and $-5M$, it is possible to use one Booth-1 encoder and one Booth-2 encoder to generate the compound coefficient, This mitigative method is called CRBBE-1.5. Suppose the binary bits b_{i+2} (b_{i+1}) and b_{i+1} b_i (b_{i-1}) are recoded to a compound coefficient $d_{i+1}d_i$ in a pair of adjacent Booth-1 and Booth-2 encoders, with the Booth-1 encoder placed before the Booth-2 encoder,. The legal duplets are shown in Table 6.14.

Table 6.14 *Permissible pairs of contiguous digits (d_{i+1} d_i) in composite Booth-1 and Booth-2 encoded number*

$d_{i+1}=1$	$d_{i+1}=0$	$d_{i+1}=\bar{0}$	$d_{i+1}=\bar{1}$
$1\bar{0}$	00	$\bar{0}\bar{0}$	$\bar{1}0$
$1\bar{1}$	01	$\bar{0}\bar{1}$	$\bar{1}1$
$1\bar{2}$	02	$\bar{0}\bar{2}$	$\bar{1}2$

Both digits of the duplets in columns $d_{i+1} = 0$ and $d_{i+1} = \bar{0}$ have the same sign. They can be easily converted to the desirable format by rewriting the most significant digit, d_{i+1} from 0 to $\bar{0}$, or vice versa, as what have been done in CRBBE-1. After which, all duplets will have the desirable positive-negative or negative-positive coding, as shown in Table 6.15.

Table 6.15 *Proposed Covalent RB Booth-1.5 encoded duplets (d_{i+1} d_i)*

$d_{i+1}=1$	$d_{i+1}=0$	$d_{i+1}=\bar{0}$	$d_{i+1}=\bar{1}$
$1\bar{0}$	$00=\bar{0}0$	$\bar{0}\bar{0}=0\bar{0}$	$\bar{1}0$
$1\bar{1}$	$01=\bar{0}1$	$\bar{0}\bar{1}=0\bar{1}$	$\bar{1}1$
$1\bar{2}$	$02=\bar{0}2$	$\bar{0}\bar{2}=0\bar{2}$	$\bar{1}2$

Therefore, the swapping criterion of the compound coefficient is given by (6.18), where the sign of the least significant digit d_i is used to identify the positive-negative or negative-positive format.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

$$(r_i^+, r_i^-) = \begin{cases} (4|d_{i+1}|, |d_i|) & \text{if } d_i \text{ is negative} \\ (|d_i|, 4|d_{i+1}|) & \text{if } d_i \text{ is positive} \end{cases} \quad (6.18)$$

Like the Normal Booth Encoder, Redundant Binary Partial Product Generator (NBE-RBPPG), the proposed Covalent Redundant Binary Booth Encoder (CRBBE) algorithm also composes every two NB partial products to one RB partial product, and generates the RB partial products with even higher efficiency than that of the NBE-RBPPG method. For a Booth-N multiplier of word length K, the number of RB partial products generated using CRBBE is:

$$\left\lceil \frac{\lceil K/N \rceil}{2} \right\rceil.$$

which is in so far the best of all methods discussed in this chapter. It is thankful that the proposed algorithm does not generate any additional correction vector to the summing tree. Therefore, lesser hardware and shorter worst case delay are assured.

6.5 Porting of CRBBE algorithm to RB multiplier architecture

The RB multipliers in this chapter are implemented with the positive-negative coding, which is easier for hardware design.

6.5.1 Circuit implementation of CRBBE encoders and RB PPGs

According to the proposed Covalent Booth encoding algorithm, the RB partial product generators are only responsible for the generation of the absolute values of the needed multiples, such as $1M$ or $0M$ in Booth-1 encoding according to the compound coefficients $r = (r^+, r^-)$. Therefore, no additional hardware is required to realize the reformatting from 0 to $\bar{0}$, or vise versa. Tables 6.11, 6.13 and 6.15, and Equations (6.16), (6.17) and (6.18) provide the hints on how to design the CRBBE encoders and their RB partial product generators.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

In CRBBE-1 algorithm, the sign of the lower significance digit d_i is responsible for whether or not to negate the absolute value of the compound coefficient. If it is negative, the generated RB partial product is kept in its original format. If it is positive, the digits r^+ and r^- of the compound coefficient (r^+, r^-) are swapped to obtain the negative RB partial product. Fig. 6.8 shows the circuit of a slice of CRBBE-1 which consists of two normal Booth-1 encoders to cogenerate the Booth encoded digits, $d_{i+1}d_i$ to form the compound coefficient (r^+_i, r^-_i). The circuit in each dash rectangle is an independent Booth-1 encoder. The active low swap flag from the common input bits of the two encoders signals the corresponding partial product generator circuit to negate the resultant digit of the RB partial product by swapping the coefficients if it is 0. No swapping is required if it is 1.

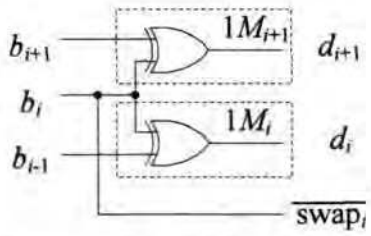


Figure 6.8 *CBEBE-1 encoder*

Fig. 6.9 shows one digit slice of the partial product generator working together with the CRBBE-1 encoder of Fig. 6.8, where the control signals $1M_{i+1}$ and $1M_i$ come from d_{i+1} and d_i, respectively, and y_{k-1}, y_k are the multiplicand bits. The swapping logic is implemented in CMOS logic style. It is also possible to be implemented in pass transistor logic.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

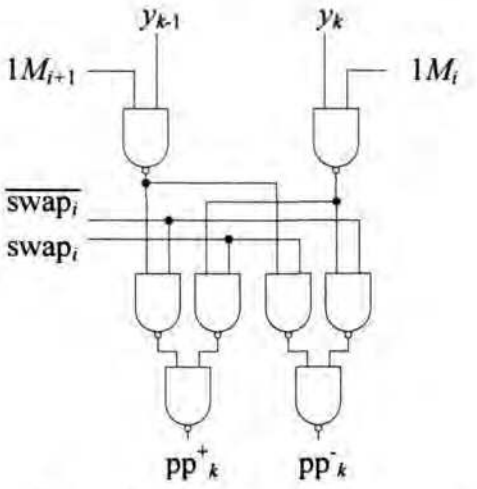


Figure 6.9 *CRBBE-1 RB partial product generator*

Fig. 6.10 shows the CRBBE-1.5 encoder, which includes a Booth-1 encoder in the upper dash box to generate the coefficient, r^+_i and a Booth-2 encoder in the lower dash box to generate the coefficient, r^-_i. The active low swap flag from the MSB input of the Booth-2 encoder signals to the corresponding partial product generator on when to swap the generated coding bits for the RB digit of the partial product.

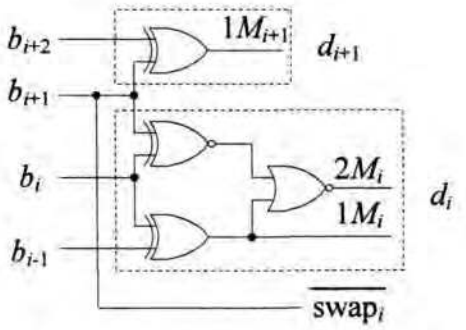


Figure 6.10 *CRBBE-1.5 encoder*

Fig. 6.11 shows one digit slice of the RB partial product generator for CRBBE-1.5 encoder.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

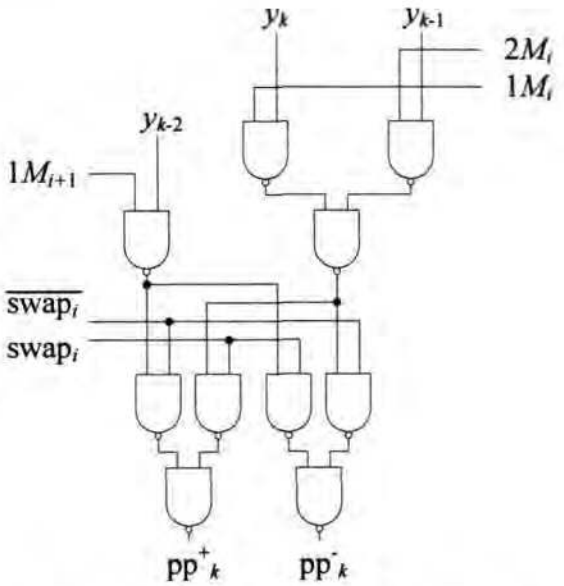


Figure 6.11 *CRBBE-1.5 RB partial product generator*

The algorithm of CRBBE-2 requires the recoding of some Booth encoded digit pairs to the desirable positive-negative pairs. The conversion function is implemented in the CRBBE-2 encoders, making the encoder more complex. However, the number of Booth encoders in a multiplier is limited, occupying only a small fraction of the total area. The excess hardware in CRBBE-2 encoders is negligible compared with the amount of hardware required in the partial product summing tree. Thus the same RB partial product generators that are used repetitively can be simplified as much as possible. Fig. 6.12(a) shows the CRBBE encoder circuit which is composed of two adjacent Booth-2 encoders. The lower encoder is fed from the binary bits $b_{2i+1}b_{2i}b_{2i-1}$ of the multiplier, and generates the signals $1m_i$, $2m_i$ and a sign bit, sgn_i taken directly from the MSB, b_{2i+1} while the upper encoder is fed from the bits $b_{2i+3}b_{2i+2}b_{2i+1}$, and generates the signals $1m_{i+1}$, $2m_{i+1}$ and a sign bit, $sgn_{i+1} = b_{2i+3}$. All these output signals will be reformatted according to Table 6.13 before they are passed to the RB partial product generators.

The reformatting circuit is shown in Fig. 6.12(b). From Table 6.13, when the most significant digit is zero, its sign bit is complemented before it is used as an active high swap flag. Otherwise, the original sign is used as an active high swap flag. An active low

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

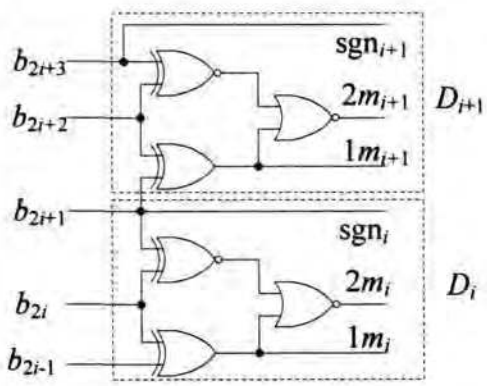
swap flag is also generated for the swapping circuit of the RB partial product generator. Therefore, we have (6.18)

$$\begin{aligned} \text{swap}_i &= (1m_{i+1} + 2m_{i+1})\,\text{sgn}_{i+1} + \overline{(1m_{i+1} + 2m_{i+1})\text{sgn}_{i+1}} \\ &= (1m_{i+1} + 2m_{i+1}) \odot \text{sgn}_{i+1} \\ &= \overline{(1m_{i+1} + 2m_{i+1})} \oplus \text{sgn}_{i+1} \end{aligned} \tag{6.18}$$

The $5M$ signal is generated by (6.19)

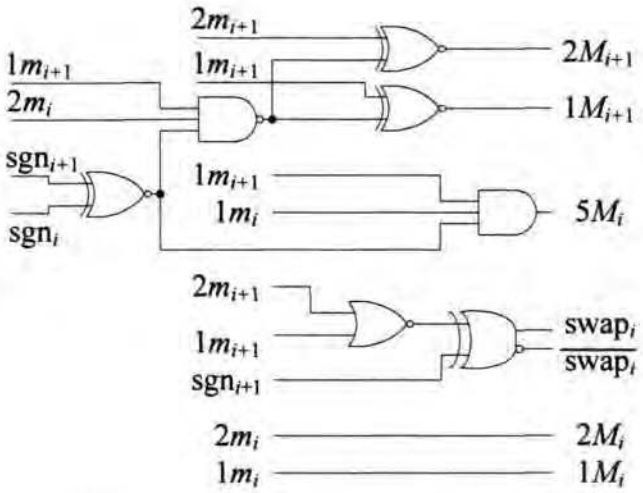
$$5M = \left(\text{sgn}_i \odot \text{sgn}_{i+1}\right) \cdot 1m_{i+1} \cdot 1m_i \tag{6.19}$$

It is noted that for each encoder the output signals $2m$ and $1m$ are mutually exclusive. The signal $5M$ is also mutually exclusive to $2m$ but not to $1m$. Such relationship can be exploited to simplify the hardware complexity of the reformatting circuits.



(a) *Booth-2 encoders for CRBBE-2*

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library



(b) *Reformatting circuits for CRBBE-2*

Figure 6.12 *CRBBE-2 encoders*

To convert the duplets 12 to $2\bar{2}$ or $\bar{1}\bar{2}$ to $\bar{2}2$, the output signals of $2m_{i+1}$ and $1m_{i+1}$ from the upper encoder are to be complemented in order to convert |1| to |2| when the contiguous digits are of the same sign and when both signals $1m_{i+1}$ and $2m_i$ are active. For all other duplets, the output signals $2m_{i+1}$ and $1m_{i+1}$ retain their original values. The converted signals, $2M_{i+1}$ and $1M_{i+1}$ of the upper encoder are given by (6.20) and (6.21):

$$\begin{aligned} 2M_{i+1} &= \left(1m_{i+1}\cdot 2m_i\cdot\left(\text{sgn}_i\odot\text{sgn}_{i+1}\right)\right)\cdot\overline{2m_{i+1}} + \overline{\left(1m_{i+1}\cdot 2m_i\cdot\left(\text{sgn}_i\odot\text{sgn}_{i+1}\right)\right)}\cdot 2m_{i+1} \\ &= \left(1m_{i+1}\cdot 2m_i\cdot\left(\text{sgn}_i\odot\text{sgn}_{i+1}\right)\right)\oplus 2m_{i+1} \\ &= \overline{\left(1m_{i+1}\cdot 2m_i\cdot\left(\text{sgn}_i\odot\text{sgn}_{i+1}\right)\right)}\odot 2m_{i+1} \end{aligned} \tag{6.20}$$

$$\begin{aligned} 1M_{i+1} &= \left(1m_{i+1}\cdot 2m_i\cdot\left(\text{sgn}_i\odot\text{sgn}_{i+1}\right)\right)\cdot\overline{1m_{i+1}} + \overline{\left(1m_{i+1}\cdot 2m_i\cdot\left(\text{sgn}_i\odot\text{sgn}_{i+1}\right)\right)}\cdot 1m_{i+1} \\ &= \left(1m_{i+1}\cdot 2m_i\cdot\left(\text{sgn}_i\odot\text{sgn}_{i+1}\right)\right)\oplus 1m_{i+1} \\ &= \overline{\left(1m_{i+1}\cdot 2m_i\cdot\left(\text{sgn}_i\odot\text{sgn}_{i+1}\right)\right)}\odot 1m_{i+1} \end{aligned} \tag{6.21}$$

In the partial product generation circuit, the hard multiple $5M$ is to be generated separately from the other multiples using the circuit of Fig. 6.4. Fig. 6.13 shows one digit of the RB partial product generator for the CRBBE-2 algorithm. The precomputed $5M$

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

multiple bits are gated by the control signal, $5M$ from the encoders. The $1M$ multiplicand bits are controlled by the output $1M$ (the control signals $1M_i$ and $1M_{i+1}$ are collectively called the $1M$ signal) of both encoders as well as the control signal, $\overline{5M}$ because the control signals $1M$ and $5M$ are not mutually exclusive. Therefore, the RB partial product is $1M$ only when the control signal, $5M$ is inactive. Control signal, $2M$ (comprising the control signals, $2M_i$ and $2M_{i+1}$) is responsible for generating the multiple $2M$ since it is mutually exclusive to $1M$ and $5M$. The logic gates in the input stage of the CRBBE-2 RB partial product generator are realized with complementary CMOS logic style to reduce their loading to the CRBBE-2 encoders because a large number of such input stages are connected in parallel and driven by the outputs of the CRBBE-2 encoders. Hence, strong driving capability is required from the CRBBE-2 encoders. The output stage of the RB partial product generator is implemented in transmission gate style with unity fanout. It drives the CMOS input stage of the redundant binary adder (RBA) in the partial product summing tree.

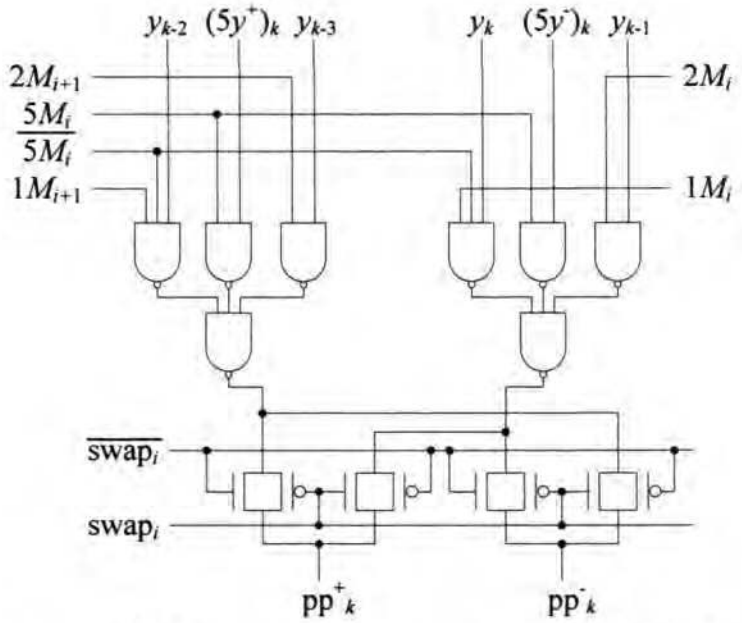


Figure 6.13 *RB partial product generator for CRBBE-2*

Compared with the RBSD multiplier, the CRBBE encoder and its corresponding RB PPG are much simpler than that of the RBSD multiplier. A legitimate comparison for the same

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

number of RB partial products generated should be Booth-2 CRBBE and RB PPG versus Booth-4 RBSD encoder and RB PPG, and Booth-1.5 CRBBE and RB PPG versus Booth-3 RBSD encoder and RB PPG. The CRBBE encoder and the RB PPG circuits are only a little more complex than the NBE-RBPPG encoder of the same radix. This small overhead is compensated by the simplified RB summing tree due to the elimination of correction vector. In some cases, it can even save a layer of RBAs.

6.5.2 Redundant Binary Adders

Redundant binary adder (RBA) is the most frequently used component in an RB multiplier, which adds two RB digits and output one RB digit. The compression ratio of RBA is 2:1. Thus, the RBA acts like a 4-2 compressor in NB multiplier. Besides, RBA summing tree has simpler interconnects than the conventional NB Wallace tree. This structural regularity can be translated into pronounced power-delay and area efficiency in deep-submicron design.

There are six possible input combinations for an RBA, which adds two RB digits, a_i and b_i and generates the intermediate sum s_i and intermediate carry c_i before it outputs the final sum d_i. The carry-free adding rule for RBA is summarized as follows. In an array of RBAs that adds two RB numbers to one, the i-th RBA computes the i-th digits, a_i and b_i of two input numbers. To avoid the carry propagation, the RBA prompts its adjacent RBA at the next significant digit position through the signal h_i. The hinting flag, h_i outputs 0 when the two input digits are both non-negative, otherwise it outputs 1. A hint of 0 indicates that the current addition will probably produce a carry of 1 while a hint of 1 means that the current addition will probably produce a carry of $\bar{1}$. The hinting flag is generated according to (6.22). The cases of "0 + 0" and "1 + $\bar{1}$" can be classified under either $h_i = 0$ or $h_i = 1$. Their classification in (6.22) is merely for the ease of hardware implementation.

$$h_i = \begin{cases} 0 & \text{if addition} \in \{0+0,\ 0+1,\ 1+1\} \\ 1 & \text{if addition} \in \{1+\bar{1},\ 0+\bar{1},\ \bar{1}+\bar{1}\} \end{cases} \tag{6.22}$$

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

The i-th RBA also receives h_{i-1} from its neighboring RBA, i.e., the $(i-1)$-th RBA, to determine its intermediate sum s_i and carry c_i. If the i-th RBA receives $h_{i-1} = 0$ from the $(i-1)$-th RBA that hinting the current addition may cause the sum to be 1, it will generate an intermediate sum s_i of $\bar{1}$ and an intermediate carry c_i of 1. Similarly, if it receives the hint $h_{i-1} = 1$ from the $(i-1)$-th RBA that the current addition may cause the sum to be $\bar{1}$, an intermediate sum s_i of 1 and an intermediate carry c_i of $\bar{1}$ will be generated. If the current addition causes the sum to be 0, the hint h_{i-1} is immaterial. The final sum d_i is obtained by adding the current immediate sum s_i and the immediate carry c_{i-1} from the $(i-1)$-th RBA. As the carry c_i is independent of c_{i-1}, the addition is carry free. The adding rules are listed in Table 6.16. The inputs are h_{i-1}, a_i and b_i while the outputs are h_i, c_i and s_i.

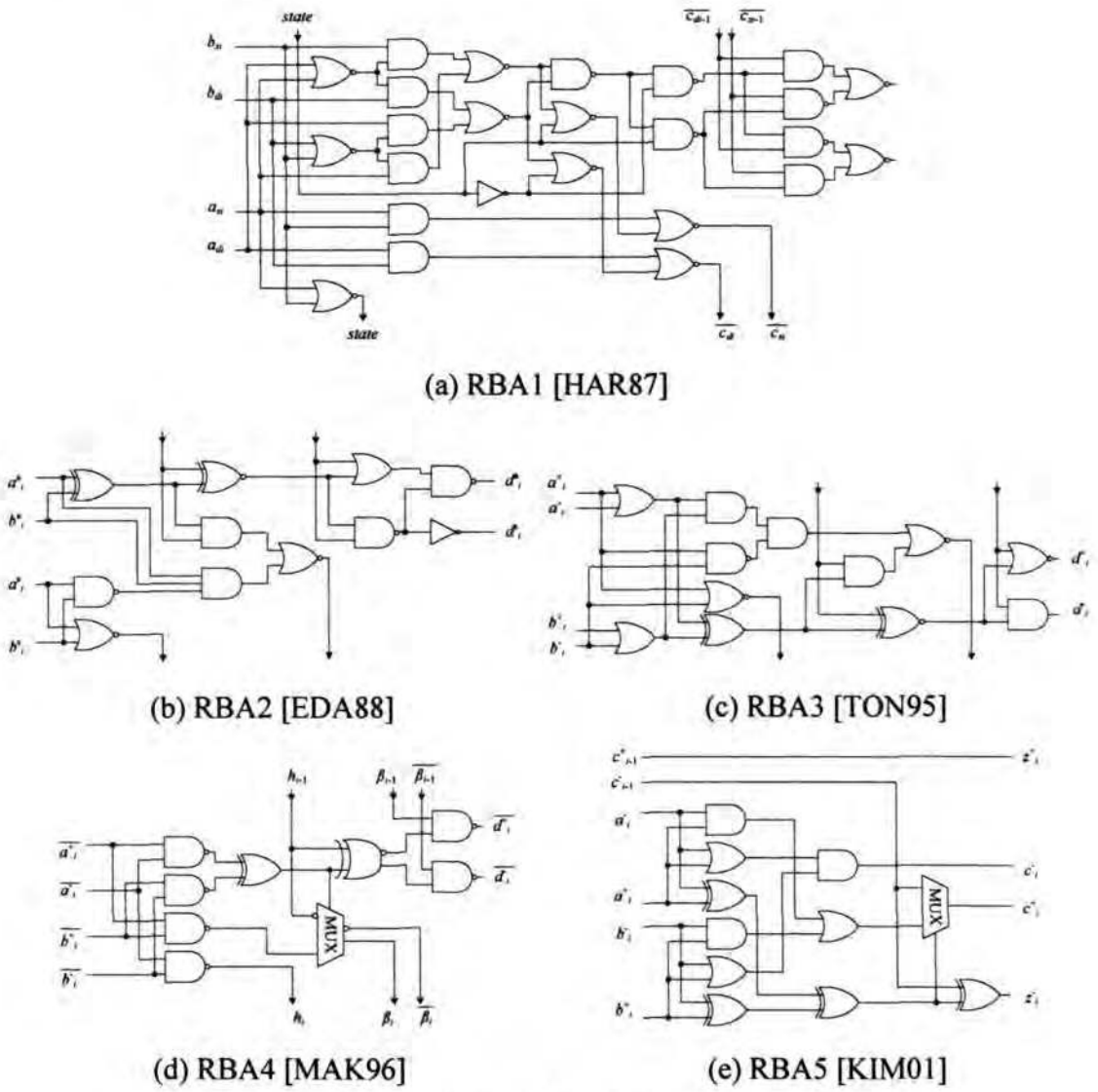
Table 6.16 *Carry-free adding rules for RBA*

a_i and b_i	h_{i-1}	h_i	c_i	s_i
0 and 0	Any	0	0	0
1 and $\bar{1}$	Any	1	0	0
0 and $\bar{1}$	0	1	0	$\bar{1}$
	1		$\bar{1}$	1
0 and 1	0	0	1	$\bar{1}$
	1		0	1
1 and 1	Any	0	1	0
$\bar{1}$ and $\bar{1}$	Any	1	$\bar{1}$	0

The RBA cells shown in Fig. 6.14 are all designed to be in compliance with the above redundant binary adding rules. The differences are that they use different coding formats and circuit optimization techniques. RBA2 uses the sign-magnitude coding for adding the two RB numbers expressed in the sign values, a_i^s and b_i^s, and the absolute values a_i^m and b_i^m to produce the sum bits, d_i^s and d_i^m. The positive-negative complement coding is adopted in RBA5. It is worth noting that this circuit is easier to optimize due to its symmetry. RBA1, RBA3 and RBA4 use the positive-negative coding. RBA1 and RBA3 use the real carry-in digit as one of the input signals while RBA4 uses the hinting flag h_{i-1} and other signals to communicate the carry-in information from the adjacent RBA of one

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

digit lower in significance. Since RBA4 is an improved version with positive-negative coding, it is used in our RB multiplier.



(a) RBA1 [HAR87]

(b) RBA2 [EDA88]

(c) RBA3 [TON95]

(d) RBA4 [MAK96]

(e) RBA5 [KIM01]

Figure 6.14 *Schematic diagrams of different RBAs*

To simplify the design of the positive-negative coding RBA, the input coding of (1,1) is refrained from feeding into the RBA directly. It is changed to (0,0) using the circuit in Fig. 6.15. This circuit is used for every digit input of the RBA summing tree to eliminate the (1,1) input.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

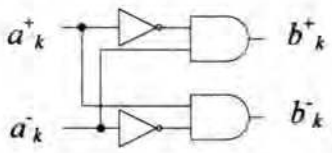


Figure 6.15 *Circuit to inhibit (1,1) input for positive-negative coding*

The circuit implementation of RBA4 is shown in Fig 6.16.

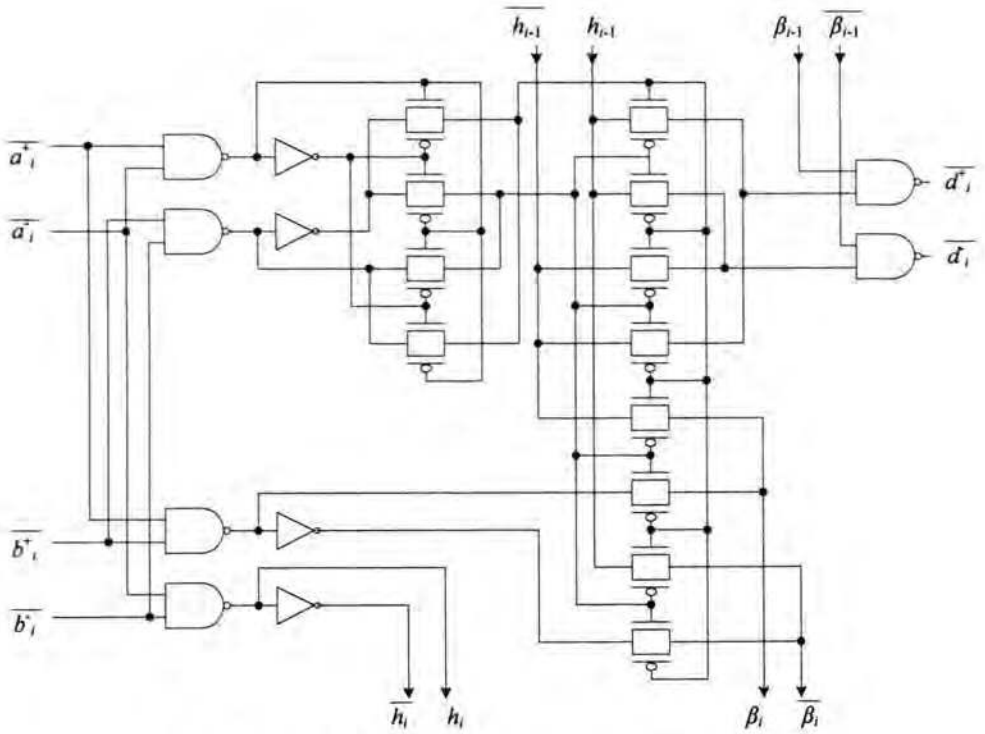


Figure 6.16 *Circuit implementation of RBA4*

6.5.3 RB to NB Conversion

The RB-to-NB conversion algorithm is derived from the basic RB number definition. Since we use the positive-negative coding, the converted value Z is shown in (6.23):

$$Z = F^+ - F^- \tag{6.23}$$

If it is to be converted to the two's complement format, as in most multipliers, we have (6.24):

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

$$Z = F^{+} + \overline{F^{-}} + 1 \tag{6.24}$$

If it is to be converted to unsigned number, as required by the IEEE floating point standard, the final result cannot be a negative number. It is equivalent to treating the F^{+} and F^{-} as two's complement numbers. Therefore, (6.24) is still valid.

The reverse conversion is optimally implemented by a carry select method [MAK96], which comprises a simple and high-speed carry propagate circuit. We denote each digit in the final RB partial product (F^{+}, F^{-}) as (f_k^{+}, f_k^{-}), and the carry-in from the next lower order digit as c_{k-1}. The k-th bit of the final sum z_k and carry-out c_k in NB format are derived in (6.25):

$$\begin{cases} z_k = f_k^{+} \oplus \overline{f_k^{-}} \oplus c_{k-1} \\ c_k = f_k^{+} \cdot \overline{f_k^{-}} + f_k^{+} \cdot c_{k-1} + \overline{f_k^{-}} \cdot c_{k-1} \\ c_0 = 1 \end{cases} \tag{6.25}$$

In the positive-negative RB coding, f_k^{+} and f_k^{-} can never become "1" simultaneously. Therefore, c_k can be rewritten as:

$$c_k = f_k^{+} \cdot \overline{c_{k-1}} + \overline{f_k^{-}} \cdot c_{k-1} \tag{6.26}$$

Using this carry select method, an N-bit converter can be implemented as shown in Fig. 6.17. Starting from one digit per group, the digits beyond the second digit are grouped in such a way that the number of digits in the group is incremented by one progressively.

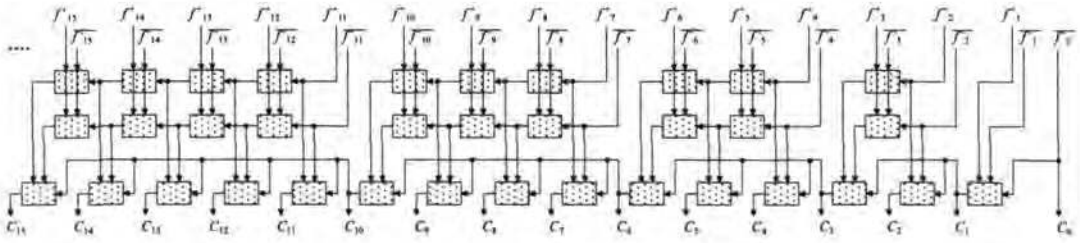


Figure 6.17 *Circuit implementation of RB-NB converter*

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

6.5.4 Proposed CRBBE 54 × 54-bit multiplier

Fig. 6.18 shows the block diagram of the proposed 54×54-bit RB multiplier architecture using the novel CRBBE-2 as the multiplier encoder. The partial product generation stage using the CRBBE-2 encoding algorithm is shown on the left hand side of the figure. In the top right hand corner is the RBA summing tree stage, while the RB-NB converter stage is on the bottom right hand corner. As mentioned before, this architecture is more hardware efficient than the other architectures to be shown later. There are 14 partial products generated directly with no correction vector. The tree structured RBA array has four levels for adding the 14 partial products into one RB number. In the RB-NB conversion stage, the final RB number with 108 digits is converted to an NB number with the same number of bits, which are accomplished in segments of different timings.

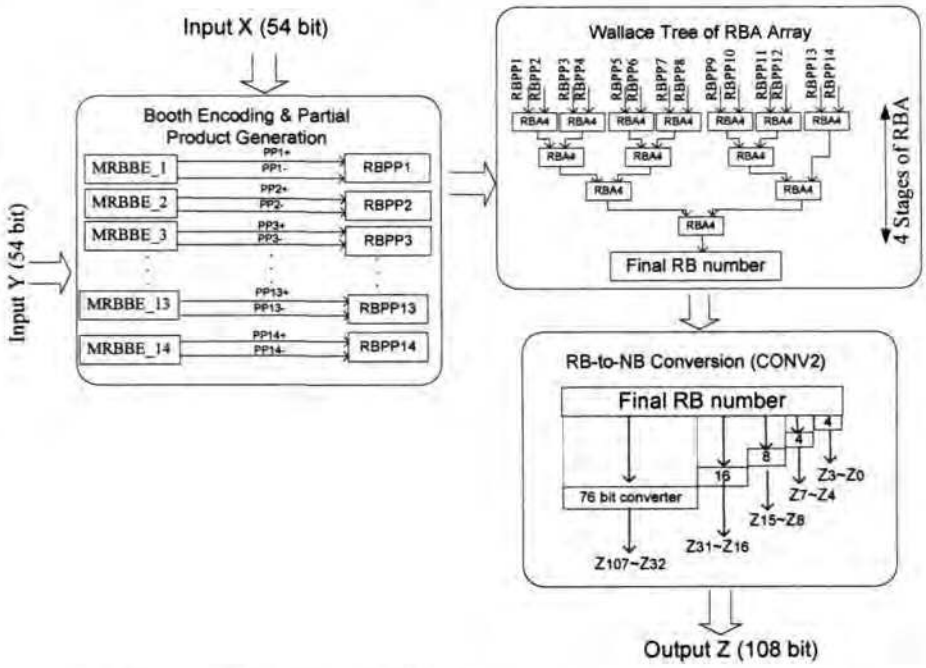


Figure 6.18 *Proposed CRBBE 54×54-bit RB multiplier*

The other architectures of 54×54-bit multipliers using the NBE-RBPPG and RBSD encoders are presented for comparison. To be fair, these architectures are chosen with the same or almost same number of RB partial products from the RB PPG. In NBE-RBPPG architecture, the partial product generation stage uses the NB Booth-2 encoder. The other

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

configurations are the same as the CRBBE architecture. Fig. 6.19 shows the block diagram of a referenced 54×54-bit NBE-RBPPG RB multiplier architecture.

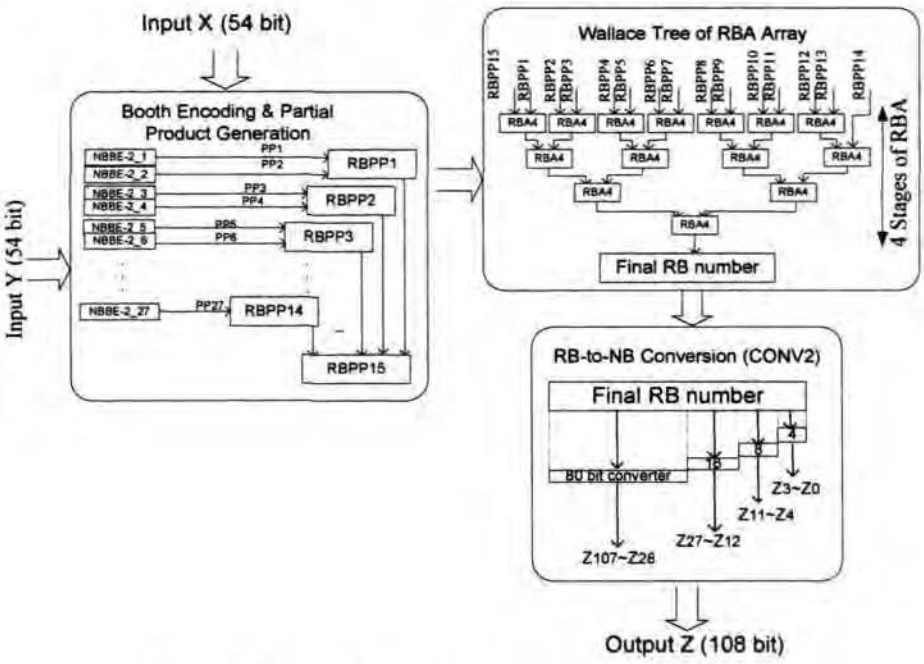


Figure 6.19 *NBE-RBPPG 54×54-bit RB multiplier*

The RBSD architecture is shown in Fig. 6.18, where Booth-4 encoder is used. Altogether 14 partial products are generated directly and no correction vector is needed. It is also noted that the RBPP14 is generated by one RB Booth-2 circuit instead of a Booth-4 circuit to reduce the amount of hardware. Fig. 6.20 shows the block diagram of the RBSD 54×54-bit RB multiplier.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

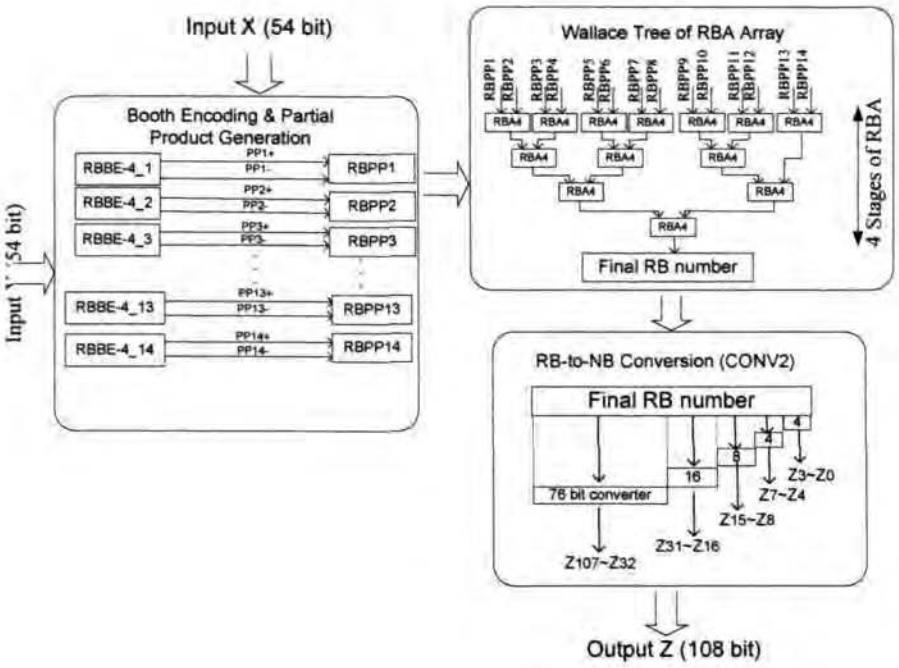


Figure 6.20 *RBSD 54×54-bit RB multiplier*

6.6 Simulation results

All the multiplier circuits stated in Section 6.5.4 are simulated in HSIM 3.0 using TSMC 0.18μm technology under the supply voltage of 1.8V. The simulation speed is set as "HSIMSPEED=0", which is the most accurate mode. 4096 randomly generated data with the input rate of 100MHz is applied to the circuits under simulation. All inputs of the multiplier are driven through buffers and all outputs are loaded with inverters to ensure that the circuits work in the same environment as in actual application. For each multiplier, the delay is calculated from the earliest transition of the 108 inputs to the latest transition of the 108 outputs. The worst case delay is the longest one of all the delays computed in the simulation process. The simulation results are shown in Table 6.17. It is evident that our proposed CRBBE multiplier outperforms the other two RB multipliers. Although the transistor count of CRBBE multiplier is a little more than that of the NBE-RBPPG multiplier, almost all these transistors are located in the input stage of the PPG circuit. This part of the circuit consumes a relatively small fraction of power of the entire multiplier because there is nearly no glitches in the PPG input stage to dissipate extra power. However, the RBA tree of the CRBBE multiplier has one row of RBAs lesser

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

than the NBE-RBPPG multiplier, which accounts for an overall power saving of 3.8%. The RBSD multiplier uses much more transistors than the other two multipliers due to its inefficient composition of PPG, which consumes an excessive number of transistors. The delay of the CRBBE multiplier is also the shortest because its PPG stage is faster than that of RBSD multiplier and its RB-NB converter has shorter critical path than that of the NBE-RBPPG multiplier. Therefore, its power delay product has a betterment of 6.3% and 10.4% over the NBE-RBPPG and RBSD multipliers, respectively.

Table 6.17 *Simulation results of RB 54×54-bit multipliers*

	Transistor count	**Power** (mW) @ 100MHz	**Delay** (ns)	**PDP** (pJ)
NBE-RBPPG	82k	26.2	4.62	121.0
RBSD	112k	27.8	4.55	126.5
CRBBE	86k	25.2	4.50	113.4

6.7 Summary

A new redundant binary multiplier is proposed based on the newly developed Covalent Redundant Binary Booth Encoding (CRBBE) algorithm. The proposed algorithm fully exploits the characteristics of the Booth encoded numbers to overcome the problem of generating hard multiples and achieves a compatible reduction of RB partial products without inducing any correction vector. Traditionally, each Booth encoded digit serves as a signed digit coefficient for the generation of a row of NB partial products and two rows of NB partial products are used to form a single RB partial product. In our proposed algorithm, adjacent coefficients form a positive-negative pair or the disqualified pair can be readily reformatted into a legal pair. Thus the RB partial products using the positive-negative coding are spontaneously generated with our CRBBE encoding. In fact, two RBSD partial products blend nicely into a RB partial product by bonding two Booth encoders covalently with an overlapping input bit to form a compound coefficient for its generation. Besides, it has the advantage that no additional RB correction vector is needed compared with the NBE-RBPPG multiplier. Thus the RBA tree is simplified with lower power dissipation and shorter worst case delay. Compared with the RBSD multiplier, which is also used to address the hard multiple problem, the proposed CRBBE

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

multiplier has only half the number of partial products to be summed if the same radix Booth encoders are used. Conversely, for the same number of RB partial products, the CRBBE encoder, say Booth-2 CRBBE, and its corresponding RB PPG is much simpler than that of the RBSD multiplier, say Booth-4 RBSD encoder and RB PPG, which uses many high fan-in gates. The simulation results show that the CRBBE multiplier consumes less power and computes faster than the NBE-RBPPG and RBSD multipliers.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Chapter 7

Conclusions and Future Research

7.1 Conclusions

In the thesis, we present several new high performance digital arithmetic cells and macrocells that focus on low power and low voltage applications, where the novelty covers from the circuit level to algorithmic level.

In the topic of full adder cell design, two novel 1-bit adder cells consisting of the XOR/XNOR, sum and carry out subcircuits, are proposed. The pass logic design style is used to efficiently generate the XOR and XNOR functions simultaneously and a good drivability carry output is generated by a novel complementary CMOS style circuit with regular layout. In addition, the last-stage inverter decouples the output and input to improve the driving capability. Despite having higher transistor count than the recently reported designs, the two cells have shown to be highly power efficient over a wide supply voltage range. The simulation results also demonstrate that the improved Module 1 of Hybrid-2 cell has made it the most power-efficient cell among a number of current-art 1-bit adder cells over a wide range of supply voltages. The energy efficiency of the Hybrid-2 cell is most pronounced at sub-1V operation. The layout of the Hybrid-2 cell shows that it is also area efficient.

An optimization procedure is proposed to size the transistors of the full adder cells in order to allow a fair comparison of different designs obtained from published literatures because the transistor sizing for optimal performance is technology dependent. The optimization procedure adopts a greedy sweeping strategy to search for the optimized

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

transistor sizes in a number of iterations until a predefined termination criterion based on the metric being optimized is reached. We have proven that the proposed algorithm is a convergent algorithm. Besides, we have also proposed a reasonably simple architecture to simulate the adder cell in an environment realistic to its actual deployment in the most frequently used parallel multiplier structure.

More complicated arithmetic building blocks have also been investigated. The architectures of 4-2 and 5-2 compressors are analyzed and different CMOS logic style circuit implementations of their constituent modules are explored. A novel 5-2 compressor architecture of 4Δ delay is proposed. In order to realistically assess and compare the figures of merits of different configurations of 4-2 and 5-2 compressors at various supply voltages, new simulation environments are established to ensure that the measured performances are still sustainable when these cells are integrated in a carry save adder (CSA) tree. The simulation results show that the 4-2 and 5-2 compressors constructed with the novel XOR* cell is able to function down to 0.6V, and features high speed and low power characteristics. Our proposed 5-2 compressor architecture outperforms all the other architectures over the range of voltages simulated, particularly when it is configured with the proposed circuits for the XOR-XNOR and the carry generator modules. Better performances against other architectures are also attained almost irrespective of the logic styles used for the circuit implementation of their constituent modules. In summary, a library of excellent power efficiency 4-2 and 5-2 compressor cells based on the advanced CMOS process technology has been developed for implementing high speed and low power multipliers operable at ultra low supply voltage.

Moving up the hierarchy of complex arithmetic circuit, a new algorithm for the design of a VLSI circuit for scalar product evaluation has been developed. The algorithm has been ported to a novel full bit parallel architecture of scalar product macrocell featuring low interconnect complexity, improved power efficiency and highly efficient VLSI area utilization. More importantly, the layout regularity and scalability enhance its performance superiority in deep submicron regime well above conventional VLSI design

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

of vector processing unit for scalar product computation. Some auxiliary circuits including a unique delay detection circuit are proposed to enable accurate delay measurement of the core under the constraint of limited IO pins. The floor-planning of the proposed architecture exploits the binary data locality through the border between the multiplication and accumulation operations based on a full combinational logic implementation. A comparison with the layout of the conventional vector multiplier shows that our proposed decomposition algorithm has led to a more compact, regular and modular physical design. A theoretical model for estimating the area and delay has been formulated. Comparing with the conventional architecture of the same capacity, the estimation shows that our design of a 16 bit scalar product multiplier on input vectors of 16 elements achieves a saving of 36.5% of silicon area, up to 59% increase in area usage efficiency and 23% decrease in interconnect wire delay. The overall performances of average power consumption, worst case delay and power efficiency of our post-layout circuit surplus even the pre-layout circuit of conventional architecture when these circuits are simulated using Synopsys Nanosim over supply voltages from 0.7V to 3.3V based on Chartered CSM 0.18μm CMOS technology. The post-layout simulation shows that the worst case delay of the core at 1.8V is 6.92ns. At 50MHz input data rate, the power dissipation is 65.0mW. The relatively small deviations between the pre- and post-layout simulation results validate the inference of the theoretical estimation that the key contributors to the delay and power reduction of our proposed architecture are the shorter and balanced global and local interconnect wires, a dominant factor in design consideration for VLSI circuits fabricated in the deep submicron technology.

In the topic of redundant binary multiplication, a new RB multiplier is proposed based on our newly developed Covalent Redundant Binary Booth Encoding (CRBBE) algorithm. The proposed algorithm fully exploits the characteristics of the Booth encoded numbers to overcome the problem of generating hard multiples and achieves a compatible reduction of RB partial products without inducing any correction vector. Traditionally, each Booth encoded digit serves as a signed digit coefficient for the generation of a row of NB partial products and two rows of NB partial products are used to form a single RB partial product. In our proposed algorithm, adjacent coefficients form a positive-negative

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

pair or the disqualified pair can be readily reformatted into a legal pair. Thus the RB partial products using the positive-negative coding are spontaneously generated with our CRBBE encoding. In fact, two RBSD partial products blend nicely into a RB partial product by bonding two Booth encoders covalently with an overlapping input bit to form a compound coefficient for its generation. Besides, it has the advantage that no additional RB correction vector is needed compared with the NBE-RBPPG multiplier. Thus the RBA tree is simplified with lower power dissipation and shorter worst case delay. Compared with the RBSD multiplier, which is also used to address the hard multiple problem, the proposed CRBBE multiplier has only half the number of partial products to be summed if the same radix Booth encoders are used. Conversely, for the same number of RB partial products, the CRBBE encoder and its corresponding RB PPG is much simpler than that of the RBSD multiplier and its RB PPG, which uses many high fan-in gates. The simulation results show that the CRBBE multiplier consumes lesser power and computes faster than the NBE-RBPPG and RBSD multipliers.

In summary, we have reported the major contributions from various levels of abstractions in VLSI design. Besides, we also presented important contributions on the algorithms and circuits for optimization, simulation and testing methods which we envision will arouse further interest in the research of low power design methodologies.

7.2 Future research

Based on the research presented in this thesis, we will now suggest several potential relevant topics for further research.

7.2.1 Exploitation of the proposed full adders in fixed multiplierless digital filter implementations

Since full adder cells are the basic building blocks of many arithmetic function units, the power-efficient Hybrid-2 cell can be used to construct those full-adder based digital systems, such as the merged arithmetic based FIR architecture proposed by our research

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

team members in [YE03, YE04]. In this architecture, the operations of the multiplication are decomposed into partial products accumulation, and merged with additions to be optimized. If these carry-save-adders to fulfill the summation of all partial products and the final carry propagate adder are implemented with our proposed Hybird-2 cell, better performance can be achieved and it would be of interest to study the level of performance that can be boosted up by our proposed full adder.

7.2.2 Exploitation of 4-2 and 5-2 compressors in the design of multiply-accumulator

Multiply-accumulator (MAC) is essentially a normal multiplication of two numbers followed by the addition of a third number to the result. In digital image and audio signal processing, the widths of the data are usually integer's power of two. The most common operand widths encountered in these multimedia applications are 8, 16 and 32. If the partial product reduction is accomplished all through the 4-2 compressors, additional layer of full adders or 4-2 compressors has to be used, which increases the delay and power consumption. With our optimized 5-2 compressor, more input terminals are available to contain the excess number of input bits at the last layer while still maintaining the same regularity of the architecture composed by only 4-2 compressors.

7.2.3 Integration of the scalar product macrocell in special purpose digital signal processors

As stated in Chapter 5, our optimized scalar product macrocell is capable of processing 16 multiplications and accumulations with the operands of 16-bit word length. It can be configured to work in either filter mode or normal scalar product multiplier mode. Therefore it is very powerful in parallel processing of digital image pixels, video and audio data stream. To maximize this capability, it can be integrated as a dedicated arithmetic unit into a single instruction multiple data (SIMD) processor or new stream based processor architecture for enhanced performance in multimedia digital processing.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

7.2.4 Design, floor planning and layout of double precision floating point multiplier with the proposed CRBBE algorithm

In IEEE 754 standard, a double precision floating point multiplication requires two 54-bit operands as multiplicand and multiplier, where 53 bits are visible by the computer users and another bit is required to ensure the rounding error is small enough not to affect the user visible bits in the final result. Thus, many proposed methods for multiplication are applied to the design of 54×54-bit multiplier. Our proposed CRBBE algorithm performs better from the prelayout simulation than other methods to realize this multiplication. It is envisaged that similar performance on real silicon can be achieved with careful design, floor planning and layout of the CRBBE double precision floating point multiplier. It would be of significant commercial value to create a fully characterized floating point RB multiplier IP core.

7.2.5 Scalable integer multiplier using the proposed CRBBE algorithm

Normally the bit-width of a digital multiplier is fixed. Therefore, the multiplier is often designed for the maximum operand width. The entire $n \times n$-bit multiplier circuit is activated to perform the operation even if the operands are less than n bits. This has severely limited the versatility of the multiplier in some applications. For example, in adaptive filter, the precision of the operands and the coefficients may change dynamically to suit the resolution of the front-end circuitry. It would be an interesting research in developing scalable multiplier, which can be reconfigured to adapt to varying input data length. Since our proposed CRBBE algorithm generates the partial product in the most efficient way without any additional correcting vector, it is a viable candidate for implementing scalable integer multiplier based on RB arithmetic. An important criterion for scalability is the ease of composing a larger word length multiplier from several smaller word length multipliers. If it is implemented with other RB multipliers, the extension of bit width is hampered by the co-generated correction vectors. If it is implemented with NB multipliers, the processing of signed number will use more multiplexers and connecting wires to select and route the partial products. Therefore, the elimination of correction vectors due to the CRBBE algorithm appear to be a promising

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

feature in fulfilling the succinctness of inter-module connectivity for the scalable multiplier.

7.2.6 Digital filters and correlators using the proposed CRBBE algorithm

One good way to realize more versatile digital filters and correlators is to use CRBBE algorithm for scalable tap multipliers as suggested in the future work Section 7.2.6. Due to the scalability offered by the algorithm, change in filter coefficients or input operand width can be addressed by conveniently reconfiguring the sizes of the filter taps. Thus the design turnaround period for similar filter order under different application environments can be greatly shortened.

7.3 Design challenges in future ultra deep submicron technologies

As technology scaling continues to advances with shrinking feature sizes, such as 90nm, 65nm, 45nm and below, it will be met by the perpetual quest for much higher performance of systems on integrated circuits. This is practical not only because higher operating frequency is easier to achieve but also due to the denser integration and low power design techniques will be much more pressing than ever. Nevertheless, there will be unprecedented challenges in leakage power and noise as the supply voltage Vdd scales down to below 1V in 65nm technology. It is estimated that at the end of 2010, the supply voltage will reach 0.6V [ITRS03]. Although the threshold voltage Vth will be proportionally scaled down to avoid the performance from being degraded exceedingly, the narrower margin between Vdd and Vth will cause more standby power and less leeway for compensation at circuit design level [MOR04]. At the same time, the interconnect wire problem arises when the designers pack more functions on a single chip with increasing bus width and higher number of global interconnect wires. The ratio of wire performance to the gate performance will worsen [HO01]. Therefore, in the sub-100nm technology era, designers will need more effort to fulfill both the demands for

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

high performance and low power. The low power and low voltage design techniques in the dissertation will continue to play an important role in the near future. To continue to benefit from the economy of scale of CMOS fabrication technology, it is also important to explore new research direction to deal with the different types of future advanced deep submicron noise such as cross talk, leakage, supply noise and process variation which are obstacles in the way of achieving the desired level of noise immunity without severely compromising the improvement already achieved in performance and energy efficiency.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Author's Publications

Journal Papers

[1] **J. Gu**, C. H. Chang and K. S. Yeo, "Algorithm and architecture of a high density, low power scalar product macrocell," *IEE Proceedings, Computers and Digital Techniques*, vol. 151, no. 2, pp. 161-172, Mar. 2004.

[2] C. H. Chang, **J. Gu** and M. Zhang, "Ultra Low Voltage, Low Power CMOS 4-2 and 5-2 Compressors for Fast Arithmetic Circuits," *IEEE Transactions on Circuits and Systems-I Regular Paper*, vol. 51, no. 10, pp. 1985-1997, Oct. 2004.

[3] C. H. Chang, **J. Gu** and M. Zhang, "Design and Analysis of Low Voltage Low Power Full Adder Cells," accepted by the *IEEE Transactions on VLSI Systems*.

Conference Papers

[1] **J. Gu**, C. H. Chang and K. S. Yeo, "An interconnect optimized floorplanning of a scalar product macrocell," in *Proc. 35th IEEE Int. Symp. on Circuits and Systems* (ISCAS'02), Scottsdale, Arizona, USA., Vol. I, pp. 465-468, May 2002.

[2] **J. Gu** and C. H. Chang, "Low voltage, low power (5:2) compressor cell for fast arithmetic circuits," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing* (ICASSP-2003), Hong Kong, Vol. II, pp. 661-664, April 2003.

[3] M. Zhang, **J. Gu** and C. H. Chang, "A novel hybrid pass logic with static CMOS output drive full-adder cell," in *Proc. 36th IEEE Int. Symp. on Circuits and Systems* (ISCAS-2003), Bangkok, Thailand, Vol. V, pp. 317-320, May 2003.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

[4] **J. Gu** and C. H. Chang, "Ultra low voltage, low power 4-2 compressor for high speed multiplications," in *Proc. 36th IEEE Int. Symp. on Circuits and Systems* (ISCAS-2003), Bangkok, Thailand, Vol. V, pp. 321-324, May 2003.

[5] C. H. Chang, M. Zhang and **J. Gu**, "A novel low power low voltage full adder cell," in *Proc. 3th IEEE/EURASIP Int. Symp. on Image and Signal Processing and Analysis* (ISPA-2003), Rome, Italy, pp. 454-458, September 2003.

[6] **J. Gu**, C. H. Chang and K. S. Yeo, "An Area and Energy Efficient IP Core for Scalar Product Computation," in *Proc. Int. Symp. on Low-Power and High-Speed Chips, Cool Chips VII*, Yokohama, Japan, Vol. 1, pp.219-226, April 14-16, 2004.

[7] C. H. Chang, Y. He and **J. Gu**, "An alternative scheme for redundant binary multiplier," in *Proc. IEEE Asia-Pacific Conf. on Circuits and Systems* (APCCAS-2004), Tainan, Taiwan, December 6-9, 2004. (Accepted for publication)

[8] Y. He, C. H. Chang, J. Gu and H. A. H. Fahmy, "A novel covalent redundant binary booth encoder," in *Proc. IEEE Int. Symp. on Circuits and Systems 2005* (ISCAS-2005), Kobe, Japan, May 23-26, 2005. (Accepted for publication)

[9] Y. He, C. H. Chang, and J. Gu, "An area efficient 64-bit square root carry-select adder for low power applications," in *Proc. IEEE Int. Symp. on Circuits and Systems 2005* (ISCAS-2005), Kobe, Japan May 23-26, 2005. (Accepted for publication)

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Bibliography

[ABU96] I. S. Abu-Khater, A. Bellaouar, and M. I. Elmastry, "Circuit techniques for CMOS low-power high-performance multipliers," *IEEE J. Solid-State Circuits*, vol. 31, pp. 1535-1546, Jun. 1996.

[ACK00] B. Ackland, and etc., "A single-chip, 1.6-billion, 16-b MAC/s multiprocessor DSP," *IEEE J. Solid-State Circuits*, vol. 35, no. 3, pp. 412-424, Mar. 2000.

[ALI02] M. Alioto and G. Palumbo, "Analysis and comparison on full adder block in submicron technology," *IEEE Trans. on VLSI Syst.*, vol. 10, no. 6, pp. 806-823, Dec. 2002.

[ASS91] W. Al-Assadi, A. P. Jayasumana and Y. K. Malaiya, "Pass-transistor logic design," *Int. J. Electronics,* vol. 70, no. 4, pp. 739-749, 1991.

[AVI61] A. Avizienis, "Signed-digit number representations for fast parallel arithmetic," *IRE Trans. on Computers,* Sep. 1961.

[BAU73] C. R. Baugh and B. A. Wooley, "A two's complement parallel array multiplication algorithm," *IEEE Trans. on Computers*, vol. 22, no. 12, pp. 1045-1047, 1973.

[BEL95] A. Bellaouar and M.I. Elmasry, *Low-Power Digital VLSI Design*. Norwell, MA: Kluwer Academic Publishers, 1995.

[BEN02] L. Benini and G. De Micheli, "Networks on chips: a new SoC paradigm," *Computer*, vol. 35, no. 1, pp. 70-78, 2002.

[BES02(1)] N. Besli, and R.G. Deshmukh, "A 54×54 bit multiplier with a new redundant binary Booth's encoding," in *Proc. IEEE Conf. on CCECE 2002*, Canada, vol. 2, pp. 597-602, May 2002.

[BES02(2)] N. Besli, and R.G. Deshmukh, "A novel redundant binary signed-digit (RBSD) Booth's encoding," in *Proc. IEEE Southeast Conf.*, pp. 427-431, Apr. 2002.

[BLA94] G. M. Blair, "Designing low-power digital CMOS," *IEEE J. Electro. & Commun. Eng,* vol. 6, pp. 229-236, Oct. 1994.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

[BON96] P. Bonatto, and V.G. Oklobdzija, "Evaluation of Booth's algorithm for implementation in parallel multipliers," in *Proc. of Twenty-Ninth Asilomar Conference on Signals, Systems and Computers,* Pacific Grove, CA, vol. 1, pp. 608-610, Nov. 1996.

[BOO51] A. D. Booth, "A signed binary multiplication technique," *Quarterly Journal of Mechanics and Applied Mathematics,* vol. 4, pp. 236-240, 1951.

[BRA63] E. L. Braun, *Digital Computer Design, Logic Circuitry, Synthesis.* Academic Press, New York, 1963.

[BRE98] L. Breveglieri and L. Dadda, "A VLSI inner product macrocell," *IEEE Trans. VLSI Syst.*, vol. 6, no. 2, pp. 292-298, 1998

[BUI00(1)] H. T. Bui, A. K. Al-Sheraidah, and Y. Wang, "New 4-transistor XOR and XNOR designs," in *Proc. of the Second IEEE Asia Pacific Conf. on ASICs*, pp. 25-28, Aug. 2000

[BUI00(2)] H. T. Bui, A. K. Al-Sheraidah, and Y. Wang, "Design and analysis of 10-transistor full adders using novel XOR-XNOR gates," in *Proc. of 5th International Conference on Signal Processing*, vol. 1, pp. 619-622, Aug. 2000

[BUI02] H. T. Bui, Y. Wang, and Y. Jiang, "Design and analysis of low-power 10-transistor full adders using novel XOR-XNOR gates," *IEEE Trans. on Circuits and Systems-II,* vol. 49, no. 1, pp. 25-30, Jan. 2002.

[CHA92] A. P. Chandrakasan, S. Sheng, and R.W.Brodersen, "Low-power CMOS digital design," *IEEE J. Solid-State Circuits,* vol. 27, pp. 473-484, Apr. 1992.

[CHA95] A. P. Chandrakasan and R. W. Brodersen, *Low Power Digital CMOS Design.* Norwell, MA: Kluwer Academic Publishers, 1995.

[CHA97] Y. Chang, J. H. Satyanarayana, and K. K. Parhi, "Low-power digit-serial multipliers," in *Proc. 1997 IEEE Int. Symp. on Circuits and Systems*, vol. 3, pp. 2164-2167, Jun. 1997.

[CHE94] Sau-Gee Chen, and Chieh-Chih Li, "Efficient designs of unified 2's complement division and square root algorithm and architecture," in *Proc. of 9th IEEE Tenconf.,* vol. 2, pp. 943 -947, 1994.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

[CHE99] K. Cheng and C. Huang, "The novel efficient design of XOR/XNOR function for adder applications," in *Proc. of IEEE Int. Conf. on Electronics, Circuits and Systems, 1999*, vol. 1, pp. 29-32, Sep. 1999

[CHO00] G. Choe and E. E. Jr. Swartzlander, "Complexity of merged two's complement multiplier-adders," in *Proc. of the 42nd Midwest Symposium on Circuits and Systems*, vol. 1, pp. 384-387, 2000.

[CHO01] I. Choo, and R. G. Deshmukh, "A novel conversion scheme from a redundant binary number to two's complement binary number for parallel architectures," in *Proc. IEEE Southeast Conf.*, pp. 196-201, Apr. 2001.

[CHO02] H. C. Chow, and I. C. Wey, "A 3.3 V 1 GHz high speed pipelined Booth multiplier," in *Proc. IEEE Int. Symp. on Circuits and Systems,* Arizona, USA, vol. 1, pp. 457-460, May 2002.

[DAD65] L. Dadda, "Some schemes for parallel multipliers," *Acta Frequenza,* vol. 45, pp. 574-480, 1965.

[DAD87] L. Dadda, "Fast serial input serial output pipelined inner product units," *Internal Rep. 87-031. Dep. Elec. Eng. Inform. Sci. Politecnico di Milano,* Milano, Italy, 1987.

[EDA88] H. Edamatsu, T. Taniguchi, T. Nishiyama, and S. Kuninobu, "A 33 MFLOPS floating point processor using redundant binary representation," in *Proc. IEEE Int. Solid-State Circuits Conference,* pp. 152-153, 342-343, Feb. 1988.

[ELR97] M. S. Elrabaa, I. S. Abu-Khater and M. Elmasry, *Advanced Low-Power Digital Circuit Techniques*. Boston: Kluwer Academic Publishers, 1997.

[ERC03] M. D. Ercegovac, T. Lang, Y. Kim, B. S. Song, J. Grosspietsch, and S. F. Gillig, "Comments on 'A carry-free 54 b x 54 b multiplier using equivalent bit conversion algorithm'," *IEEE J. Solid-State Circuits,* vol. 38, no. 1, pp. 160-161, Jan. 2003.

[FAN96] S. Fang, J. Wang, and W. Feng, "A new direct design for three-input XOR function on the transistor level," *IEEE Trans. On Circuits and Systems I,* vol. 43, no. 4, pp. 343-348, April 1996.

[FAN00] L. Fanucci and M. Forliti, "Interlaced diagonal-wise pipelined serial

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

multiplier," *Electronics Letters*, vol. 36, no. 21, pp. 1824-1825, Oct. 2000.

[FEI97] K. A. Feiste and E. E. Jr. Swartzlander, "Merged arithmetic revised," in *Proc. of the Workshop on Signal Processing Systems*, pp. 212-221, 1997.

[FER89] D. K. Ferry, L. A. Akers and E. W. Greeneich, *Ultra Large Scale Integrated Microelectronics*. Englewood Cliffs, NJ: Prentice Hall, 1989.

[GEO01] V. George and J. M. Rabaey, *Low-Energy FPGAs – Architecture and Design*. Boston: Kluwer Academic Publishers, 2001.

[GRG01] S. Grgic, M. Grgic, B. Zovko-Cihlar, "Performance analysis of image compression using wavelets," *IEEE Trans. on Industrial Electronics*, vol. 48, no. 3, pp. 682 -695, 2001.

[GU03(1)] J. Gu and C. H. Chang, "Ultra low voltage, low power 4-2 compressor for high speed multiplications," in *Proc. 36th IEEE Int. Symp. on Circuit and Systems*, Bangkok, Thailand, vol. V, pp. 321-324, May 2003.

[GU03(2)] J. Gu and C. H. Chang, "Low voltage, low power (5:2) compressor cell for fast arithmetic circuits," in Proc. *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Hong Kong, vol. II, pp. 661-664, Apr. 2003.

[GUN99] J. E. Gunn, K. Barron, and W. Ruczczyk, "A low-power DSP core-based software radio architecture," *IEEE J. on Selected Area in Communications*, vol. 17, no. 4, pp. 574-590, 1999.

[HAR87] Y. Harata, Y.Nakamura, H. Nagase, M. Takigawa, and N. Takagi, "A high-speed multiplier using a redundant binary adder tree," *IEEE J. Solid-State Circuits*, vol. 22, no. 1, pp. 28 -34, Feb. 1987.

[HAS01] Y. M. Hasan, L. J. Karam, M. Falkinburg, A. Helwig, and M. Ronning, "Canonic Signed Digit Chebyshev FIR Filter Design," *IEEE Signal Processing Letters*, vol. 8, no. 6, pp. 167-169, 2001.

[HO01] R. Ho, K. Mai, and M. Horowitz, "The Future of Wires," Proc. IEEE, Apr. 2001, pp. 490-504.

[HSI98] S. F. Hsiao, M. R. Jiang, and J. S. Yeh, "Design of high-speed low-power 3-2 counter and 4-2 compressor for fast multipliers," *Electronics Letters*, vol. 34, no. 4, pp. 341 –343, 1998.

[ITRS03] *The International Technology Roadmap for Semiconductors*, 2003.

[KAT00] S. Katkrori, and S. Alupoaei, "RT-level interconnect optimization in DSM

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

regime," in *Proc. IEEE Computer Society Workshop on VLSI,* Orlando, FL, pp. 143-148, 2000.

[KHA01] S. P. Khatri, R. K. Brayton and A. L. Sangiovanni-Vincentelli, *Cross-Talk Noise Immune VLSI Design Using Regular Layout Fabrics*. Boston: Kluwer Academic Publishers, 2001.

[KHO99] K. Khoo, Z. Yu, and Jr. A. N. Willson, "Improved-Booth encoding for low-power multipliers," in *Proc. IEEE Int. Symposium on Circuits and Systems,* San Diego, CA, USA, vol. 1, pp. 62 -65, 1999.

[KIM01] Y. Kim, B. S. Song, J. Grosspietsch, and S. F. Gillig, "A carry-free 54b x 54b multiplier using equivalent bit conversion algorithm," *IEEE J. Solid State Circuits,* vol. 36, no.10, pp. 1538 -1545, Oct. 2001.

[KIM03] Y. Kim, B. S. Song, J. Grosspietsch, and S. F. Gillig, "Correction to "a carry-free 54 b x 54 b multiplier using equivalent bit conversion algorithm"," *IEEE J. Solid-State Circuits,* vol. 38, no. 1, pp. 159-159, Jan. 2003.

[KO95] U. Ko, P. Balasara, and W. Lee, "Low-power design techniques for high-performance CMOS adders," *IEEE Trans. on VLSI Syst.,* vol. 3, no. 2, pp.327-333, Jun. 1995.

[KWO00] O. Kwon, K. Nowka, and E.E. Swartzlander, "A 16-bit x 16-bit MAC design using fast 5:2 compressor," in *Proc. IEEE Int. Conf. on Application-Specific Systems, Architectures, and Processors*, pp. 235 -243, 2000.

[LEE97(1)] T. Lee and J. Cong, "The new line in IC design," *IEEE Spectrum,* vol. 34, no. 3, pp. 52-58, 1997.

[LEE97(2)] H. Lee and G. E. Sobelman, "New low-voltage circuits for XOR and XNOR," in *Proc. IEEE Southeastcon '97, 'Engineering new New Century'*, pp. 225-229, Apr. 1997.

[LEE02] S. Lee, S. Bae, and H. Park, "A compact Radix-64 54×54 CMOS redundant binary parallel multiplier," *IEICE Trans. Electron.*, vol. E85-C, no.6, Jun. 2002.

[LIN01] R. Lin, "Reconfigurable parallel inner product processor architectures," *IEEE Trans. VLSI Syst.*, vol. 9, no. 2, pp. 261-272, 2001.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

[LU95] S. Lu and J. Kenney, "Design of most-significant-bit-first serial multiplier," *Electronics Letters*, vol. 31, no. 14, pp. 1133-1135, Jul. 1995.

[LU01] J. Lu, Y. Shu, Z. Lin, and L. Wang, "A novel 10-transistor low-power high-speed full adder cell," in *Proc. 6th Int. Conf. on Solid-State and Integrated-Circuit Technology*, vol. 2, pp. 1155-1158, Oct. 2001.

[MAC61] O. L MacSoreley, "High-speed arithmetic in binary computers," *IRE Proceedings,* vol. 49, pp. 67-91, 1961.

[MAK96] H. Makino, Y. Nakase, H. Suzuki, H. Morinaka, H. Shinohara, and K. Mashiko, "An 8.8-ns 54×54-bit multiplier with high speed redundant binary architecture," *IEEE J. Solid State Circuits,* vol. 31, no. 6, pp. 773-783, Jun. 1996.

[MAR99] M. Margala and N. G. Durdle, "Low-power low-voltage 4-2 compressors for VLSI applications," in *Proc. IEEE Alessandro Volta Memorial Workshop on Low-Power Design,* Como, Italy, pp. 84-90, 1999.

[MEH91] M. Mehta, V. Parmar, and E. Jr. Swartzlander, "High-speed multiplier design using multi-input counter and compressor circuits," in *Proc. 10th IEEE Symp. on Computer Arithmetic*, pp. 43 -50, Jun 1991.

[MOR04] E. Morifuji, and etc, "New guideline of Vdd and Vth scaling for 65nm technology and beyond," in *Proc. IEEE 2004 Symposium on VLSI Technology Digest of Technical Papers*, pp. 164-165, Jun. 2004.

[MUH01] K. Muhammas, "Speed, power, area and latency tradeoffs in adaptive FIR filtering for PRML read channels," *IEEE Trans. on VLSI Systems*, vol. 9, no. 1, pp. 42-51, 2001

[MUR96] F. Murabayashi, T. Yamauchi, H. Yamada, T. Nishiyama, K. Shimamura, S. Tanaka, T. Hotta, T. Shimizu, and H. Sawamoto, "2.5V CMOS Circuit Techniques for a 200 MHz superscalar RISC Processor," *IEEE J. Solid-State Circuits*, vol. 31, no. 7, pp. 972-980, Jul. 1996.

[MIL92] B. Millar, P. E. Madrid, and E. E. Swartzlander, "A fast hybrid multiplier combining Booth and Wallace/Dadda algorithms", in *proc. 35th Midwest Symposium on Circuits and Systems,* Washington DC, vol. 1, pp.158-165, 1992.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

[NAG96] C. Nagendra, M. J. Irwin, and R. M. Owens, "Area-time-power tradeoffs in parallel adders," *IEEE Trans. on Circuits and Systems – II: Analog and Digital Signal Processing*, vol. 43, no. 10, pp. 689-702, 1996.

[NAN99] A. Nannarelli, and T. Lang, "Low-power divider," *IEEE Trans. on Computers*, vol. 48, no. 1, pp. 2-14, 1999.

[NAY99] S. S. Nayak and P. K. Meher, "High throughput VLSI implementation of discrete orthogonal transformation using bit-level vector-matrix multiplier," *IEEE Trans. on Analog and Digital Signal Processing*, vol. 46, no. 5, pp. 655-658, 1999.

[OKA91] F. Okamoto, Y. Hagihara, C. Ohkubo, N. Nishi, H. Yamada, and T. Enomoto, "A 200-MFLOPS 100-MHz 64-b BiCMOS vector-pipelined processor (VPP) ULSI," *IEEE J. Solid-State Circuits*, vol. 26, no. 12, pp. 1885-1893, 1991.

[OKL96] V. G. Oklobdzija, D. Villeger, and S. S. Liu, "A method for speed optimized partial product reduction and generation of fast parallel multipliers using an Algorithmic Approach," *IEEE Trans. on Computers*, vol. 45, no.3, pp. 294-305, 1996.

[PAR00] B. Parhami, *Computer Arithmetic Algorithms and Hardware Designs*. OXFORD University press, New York, 2000.

[PAR97] K. K. Parhi, "Fast VLSI binary addition," *IEEE Workshop on Signal Processing Systems, SIPS 97 -Design and Implementation*, Leicester,UK, pp. 232 -241, 1997.

[PEK03] K. Z. Pekmestzi, P. Kalivas, N. Moshopoulos, and J. Sifnaios, "Complex constant number serial multipliers," *IEE Proc. Circuits, Devices and Systems*, vol. 150, no. 5, pp. 405-410, Oct. 2003.

[PIG95] C. Piguet, "Logic design for low-power CMOS circuits, " in *Proc. of 1995 IEEE Region 10 Int. Conf. on Microelectronics and VLSI*, pp. 299-302, Nov. 1995.

[PRA01] K. Prasad and K. K. Parhi, "Low-power 4-2 and 5-2 Compressors," in *Proc. 35th Asilomar Conf. on Signals, Systems and Computers*, vol.1, pp. 129-133, 2001.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

[QUI01] J. M. Quintana, M. J. Avedillo, R. Jimenez, and E. Rodriguez-Villegas, "Low-power logic styles for full-adder circuits," in *Proc. of the 8th IEEE Int. Conf. on Electronics, Circuits and Systems, 2001*, vol. 3 , pp. 1417-1420, Sept. 2001

[RAB96] J. M. Rabaey, *Low power Design Methology*. Boston: Kluwer Academic Publishers, 1996.

[RAB01] J. M. Rabaey, *Digtal Integrated Circuits-a design perspective*. Prentice Hall Press, 2001

[RAD99] D. Radhakrishnan, "Low voltage CMOS full adder cells," *Electronics Letters*, vol. 35, pp. 1792-1794, 1999.

[RAD00] D. Radhakrishnan, and A.P. Preethy, "Low power CMOS pass logic 4-2 compressor for high-speed multiplication," in *Proc. of the 43th IEEE Midwest Symp. on Circuits and Systems*, Lansing MI, vol. 3, pp. 1296-1298, 2000.

[RAD01] D. Radhakrishnan, "Low-voltage low-power CMOS full adder," *IEE Proceedings on Circuits, Devices and Systems*, vol. 148, no. 1, pp. 19-24, Feb. 2001.

[REN93] H. Ren, L.B. Hoang, H. C. Chen, and B.W.Y. Wei, "Design of a 16-bit CMOS divider/square-root circuit," in *Proc. 27th Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, vol.1, pp. 807 -811, Nov. 1993.

[RUL03] W. Rulling, "Remark on carry-free binary multiplication," *IEEE J. Solid-State Circuits*, vol. 38, no. 1, pp. 159-160, Jan. 2003.

[SAK00] H. Sakamoto, H. Ochi, K. Uda, K. Taki, Bu-Yeol Lee, and T. Tsuda, "A 16-bit redundant binary multiplier using low-power pass-transistor logic SPL," in *Proc. ASP-DAC 2000 Asia and South Pacific Design Automation Conference*, Yokohama, Japan, Jan. 2000.

[SAY02] M. Sayed and W. Badawy, "Performance analysis of single bit full adder cells using 0.18, 0.25 and 0.35um CMOS technologies,"in *Proc. Int. Symp. Circuits and Syst.*, Scottsdale, Arizona, USA, 2002.

[SHA93] N. R. Shanbhag and K. K. Parhi, "Relaxed look-ahead pipelined LMS

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

adaptive filters and their application to ADPCM coder," *IEEE Trans. Circuits Syst. II*, vol. 40, no. 12, pp. 753-766, 1993.

[SHA97] A. M. Shams, and M. A. Bayoumi, "A structured approach for designing low power adders," in *Proc. 31st Asilomar Conf. on Signals, Systems & Computers,* Pacific Grove, CA, vol. 1, pp. 757-761, 1997.

[SHA98] A. M. Shams and M. A. Bayoumi, "A novel low-power building block CMOS cell for adders," in *Proc. 1998 IEEE Int. Symp. on Circuits and Systems*, vol. 2, pp. 153-156, Jun. 1998.

[SHA99] A. Shams and M. Bayoumi, "Performance evaluation of 1-bit CMOS adder cells," in *Proc. of IEEE Int. Symp. on Circuit and Systems*, pp.27-30, Jul. 1999.

[SHA00] A. M. Shams and M. A. Bayoumi, "A novel high-performance CMOS 1-bit full-adder cell," *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 47, no. 5, pp. 478-481, 2000.

[SHA02] A.M. Shams, T.K. Darwish, and M.A. Bayoumi, "Performance analysis of low-power 1-bit CMOS full adder cells," *IEEE Trans. on VLSI Syst.*, vol. 10, no. 1, pp. 20 -29, Feb. 2002.

[SHI91] F. W. Shih, "High performance self-checking adder for VLSI processor," in *Proc. IEEE Custom Integrated Circuits Conference*, pp. 15.7/1-15.7/3, May 1991.

[SHI97] K. W. Shin, and B. S. Song, "A complex multiplier architecture based on redundant binary arithmetic," in *Proc. IEEE Int. Symp. on Circuits and Systems,* Hong Kong, vol. 3, pp. 1944 – 1947, Jun. 1997.

[SON91] P. J. Song, and G. De Micheli, "Circuit and architecture trade-off for high-speed multiplication," *IEEE J. Solid-State Circuits,* vol. 26, pp.1184 –1198, 1991.

[SRI92] H.R. Srinivas, and K.K. Parhi, "A fast VLSI adder architecture," *IEEE J. Solid-State Circuits,* vol: 27 no. 5, pp.761 -767, May 1992.

[SWA80] E. E. Jr. Swartzlander, "Merged Arithmetic," *IEEE Trans. on Computers*, vol. c-29, pp. 946-950, 1980.

[SYL98] D. Sylvester, and K. Kuetzer, "Getting to the bottom of deep submicron," in

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Proc. Int. Conf. on Computer-Aided Design, San Jose, CA, pp. 203–211, 1998.

[TAK85] Takagi N., Yasuura H., and Yajima S., "High-speed VLSI multiplication algorithm with a redundant binary addition tree," *IEEE Trans. on Computers*, pp. 789-796, 1985.

[TOC98] R. J. Tocci and N. S. Widmer, *Digital systems: Principle and Application.* 7th Edition, Prentice Hall Inc., 1998.

[TON95] M. Tonomura, "High-speed digital circuit of discrete cosine transform," *IEICE Trans. Fundamentals,* vol. E78-A, no. 8, Aug. 1995.

[TSA02] T. K. K. Tsang and M. N. El-Gamal, "Gain and frequency controlled sub-1V 5.8 GHz CMOS LNA," in *Proc. IEEE Int. Symp. on Circuits and Systems*, vol. 4, pp. 795-798, 2002.

[VAI01] M. M. Vai, *VLSI Design*. Boca Raton: CRC Press, 2001.

[VES99(1)] M. Vesterbacka, "A 14-transistor CMOS full adder with full voltage-swing nodes," in *Proc. 1999 IEEE Workshop on Signal Processing Systems*, pp. 713-722, Oct. 1999

[VES99(2)] M. Vesterbacka, "A new six-transistor CMOS XOR circuit with complementary output," in *Proc. 42nd Midwest Symp. on Circuits and Systems*, vol. 2, pp. 796-799, Aug. 1999.

[WAL64] C. S. Wallace, "A suggestion for fast multipliers," *IEEE Trans. on Comput.,* vol. 13, pp. 14-17, 1964.

[WAL00] M. G. Walker, "Modeling the wiring of deep submicron ICs," *IEEE Spectrum,* pp. 65-71, Mar. 2000.

[WAN94] J. Wang, S. Fang, and W. Feng, "New efficient designs for XOR and XNOR functions on the transistor level," *IEEE J. Solid-State Circuits*, vol 29, no. 7, pp. 780-786, Jul. 1994.

[WAN95] Z. Wang, G. Jullien, W. C. Miller, "A new design technique for column compression multipliers," *IEEE Trans. on Comput.,* vol. 44, no. 8, pp. 962-970, Aug. 1995.

[WEI81] A. Weinberger, "4-2 carry-save adder module," *IBM Tech. Disclosure Bulletin,* vol. 23, 1981.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

[WES93] N. Weste and K. Eshraghian, *Principles of CMOS VLSI Design, A System Perspective*. Reading, MA: Addison-Wesley, 1993.

[WEY02] I. Wey, C. Huang, and H. Chow, "A new low-voltage CMOS 1-bit full adder for high performance applications," in *Proc. 2002 IEEE Asia-Pacific Conference on ASIC*, pp. 21-24, Aug. 2002.

[WOL98] W. Wolf, *Modern VLSI Design, Systems on Silicon*. Second Edition, Upper Saddle River, NJ: Prentice Hall, 1998.

[WON95] C. S. H. Wong, J.C. Rudell, G. Uehara, and P. R. Gray, "A 50 MHz eight-tap adaptive equalizer for partial-response channels," *IEEE J. Solid State Circuits*, vol. 30, no. 3, pp. 228-234, 1995.

[YEO02] K. S. Yeo, S. S. Rofail and W. L. Goh, *CMOS/BiCMOS ULSI, Low Voltage, Low Power*. Upper Saddle River, NJ: Prentice Hall, 2002.

[YU00] C. C. Yu, W. P. Wang, and B. D. Liu, "A 3-input XOR/XNOR for low-voltage low-power applications," on *Proc. of the 2000 IEEE Asia-Pacific Conf. on Circuits and Systems*, pp. 505-508, Dec. 2000.

[YU02] Z. Yu, M. L. Yu, K. Azader and A. N. Willson Jr., "A low power adaptive filter using dynamic reduced 2's complement representation," in *Proc. IEEE Custom Integrated Circuit Conf.*, Orlando, FL, pp. 141-144, 2002.

[ZHA03] M. Zhang, J. Gu and C. H. Chang, "A novel hybrid pass logic with static CMOS output drive full-adder cell," in *Proc. 36th IEEE Int. Symp. on Circuits and Systems*, Bangkok, Thailand, Vol. V, pp. 317-320, May 2003.

[ZHU92] N. Zhuang and H. Hu, "A new design of the CMOS full adder," *IEEE J. Solid-State Circuits*, vol. 27, no. 5, pp. 840-844, May 1992.

[ZIM97] R. Zimmermann and W. Fichtner, "Low-power logic styles: CMOS versus pass-transistor logic," *IEEE J. Solid-State Circuits*, vol. 32, no. 7, pp. 1079-1090, Jul. 1997.